World RetroMagazine
future days are back
ISSUE 13 FEBRUARY 2022 - WWW.RETROMAGAZINE.NET - FREE PUBLISHING
ISSN 3103-697X

A lot of

REVIEWS

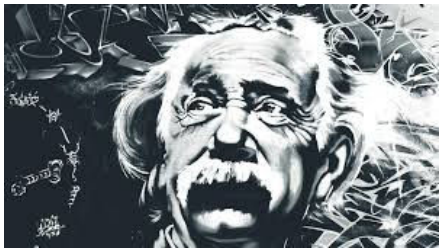
for many game systems

Konami on AMIGA!!!

HARDWARE REVIEW
FAIRCHILD CHANNEL F

Hardware: ANALOGUE POCKET: the ultimate portable - GAME BOY: The (S)protection
Sections: R-TYPE Shoot em up YEAR ZERO - A BIT OF RARITY: CREATING Q*BERT
Programming: FOCAL LANGUAGE first part - FRACTAL IN BASIC third part
RETRO PROGRAMMING ITALIA: we program Basic - Master Music SID
Remembering JERRY LAWSON
... and so much more!!!

Necessity is the Mother of invention



RetroMagazine World, with every issue in two versions, Italian and English, registers an ever-growing number of readers and arouses constant interest in the topics we cover. Our choice to avoid the distribution of printed copies, in addition to keeping us away from a number of problems arising from the management of subscriptions and shippings, brings some advantages, not least of which is that we can continue to guarantee the complete gratuity of the magazine, and allows us to reach anywhere in the world with a simple URL. The number of downloads of the new issues as well as of the previous ones testifies the goodness of this decision. Moreover, the Common Creative license used to release the magazine does not prohibit printing hard copies and, in fact, we are aware of the many retrocomputing groups that produce bound copies of issues and make them available to all members of their clubs.

Another difference of RMW from the "classic" magazines of the fantastic 80's and 90's is the lack of a permanent column hosting Readers' Mail. One of the most important reasons why RMW keeps being released is the preservation of the knowledge and memory of all the home computing systems in history, so we give a lot of importance to feedback and interaction with all the old and new retrocomputing enthusiasts. But the scenario (if compared to the early years of 8/16 bit machines diffusion) has totally changed. In the past, newsstand magazines were basically the only dynamic medium for computer news. The editorial offices were by necessity the point of reference for all those approaching the world of computers and game consoles for the first time. Often the mail columns were targeted by questions of all kinds and in some cases, rather than serving as a centre of generic information, became a battleground between "fans" of different platforms.

Nowadays, we have dozens of possible tools for interaction available (from simple e-mail to the most advanced instant messaging tools) but the feedback we receive from readers regarding discussions and insights generated by our articles and reviews is quite small. We at RMW run a Facebook page, an updated website, a Twitter account, Instagram, Telegram, YouTube, etc., and we feel it is our duty to open and manage the flow of information using all these channels for anyone to interact with us, but, apart from a few isolated cases, reader comments are often concentrated around a few posts published on the FB page. Too little, especially if we think that retrocomputing in the last 4-5 years has “infected”, please forgive the use of this term, hundreds of thousands of fans, even those belonging to younger generations.

So what has changed for readers when they browse a magazine like RMW, albeit it is only digitally published? Have we impoverished ourselves in our ability to interact? Have we become lazier and more inclined to “get lost” in the vast amount of information that the Net continually provides us, although in the form of pills? Do we prefer a less structured and rigorous kind of information today, based more on word of mouth and the experience of individuals on social media and digital forums? Could it be that 40 years ago, the lack of tools and documentation increased our desire for documentation and communicative exchange with many interlocutors? Could it be that only "hunger" drives inventiveness and aids creativity and discussion?

David La Monaca

TABLE OF CONTENT

◊ GameBoy & GB Color - the (un)protections Pag. 3
◊ ZX99 and other unofficial add-ons... Pag. 8
◊ Analogue Pocket... Pag. 10
◊ Fairchild Channel F Pag. 12
◊ FOCAL language on PDP-8 - part 1 Pag. 18
◊ 8-Bit fractals in BASIC sauce - part 3 Pag. 22
◊ C128 in 80 column mode - part 2 Pag. 26
◊ How to create a BASIC game for C64 - p1 Pag. 28
◊ Musica Maestro SID! Pag. 32
◊ Interview with Bas Scheijde... Pag. 35
◊ Interview with Alessandro Rogati... Pag. 38
◊ Remembering Jerry Lawson... Pag. 42
◊ Warren Davis and the invention of Q*bert Pag. 44
◊ Sometimes they come back... Pag. 45
◊ R-TYPE - Shoot'em up year zero Pag. 46
◊ The Age of Heroes (C64) Pag. 50
◊ The Shadows of Sergoth (Amiga) Pag. 52
◊ Puzzle Bobble (C64) Pag. 54
◊ Wizard ofethe Board (N64) Pag. 56
◊ Silver Valley (SMS) Pag. 58
◊ Knightmare (Amiga) Pag. 60
◊ Retaliate C.E. (C64) Pag. 62
◊ Mega Turrican Director's Cut (MegaDrive) Pag. 64
◊ Eyra, the Crow Maiden (MegaDrive) Pag. 66
◊ Knight Guy in Low Res World (Atari 7800) Pag. 68
◊ Robot in Action (C64) Pag. 69
◊ Shonen Jump's One Piece (GameBoy) Pag. 70
◊ Gremlins 2 the New Batch (GameBoy) Pag. 71
◊ Crisis Force (NES) Pag. 72
◊ Classic Kong (SNES) Pag. 74
◊ Prohibition (C64) Pag. 75
◊ Retro Multiplayer Pag. 76

People involved in the preparation of this issue of RetroMagazine World (in no particular order):

- Alberto Apostolo
- Dr. Andrea Q.
- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- Mic the Biker Novarina
- Francesco Fiorentini
- David La Monaca
- Leonardo Miliani
- Barbara “Morgana” Murgida
- Takahiro Yoshioka
- Roberto Del Mar Pirazzini
- Ingrid Poggiali
- Gianluca Girelli
- Giampaolo Moraschi
- Querino Ialongo
- Giorgio Balestrieri
- Felice Nardella
- Franco Bressan
- Cover Image: **Giuseppe Mangini**
- Cover layout: **Carlo N. Del Mar Pirazzini**

GameBoy and GameBoy Color - the (un)protections

by Dr. Andrea Q. - www.retrofixer.it

Youtube: https://www.youtube.com/channel/UCEw0CQ8LKyA9jVvWXkEwp4Q

You who don't know what a Game Boy is, raise your hand! Anyone? Well, I was sure of it! Just as I'm sure that the Game Boy Color is in everyone's memory.

However, not everyone knows that its code name was DMG-01 (1989-1990), whilst that of the Game Boy Color (1998) was CGB-01.
The first "small" and portable Nintendo devices are certainly among the most famous gems ever released in the world of consoles. And they are even not less than their "big sisters" when it comes to the protection systems.

The Game Boy is in fact equipped with a BootROM that we also find in the Game Boy Color (GBC) and Super Game Boy (which is a cartridge-adapter to play GB games on Super Nintendo).

THE PROTECTION OF THE FIRST NINTENDO LAPTOPS

The BootROM with which the GB, GBC and SGB hardware is equipped is 256 bytes in size and has 3 special "protective" features:

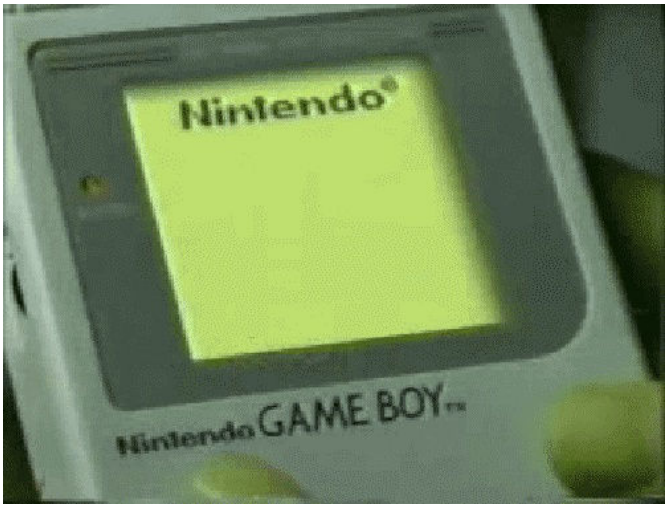
1 - The BootROM itself is excluded from reading with the last instruction of its code: this "lock out" was perhaps not intended, but was probably necessary for reasons related to (low) system resources. The BootROM is mapped in memory between offset 0x0000 and offset 0x00FF (256 bytes). The last 2 bytes represent an instruction that writes to a special register blocking that memory address from being read, so that all subsequent instructions (which are, starting from offset 0x0100, those contained in the cartridge) cannot read it;

2 - inside it contains code that reads the 48 bytes of the NINTENDO logo present in the cartridge (offsets ranging from 0x104 to 0x133 of each cartridge):

and shows them on the screen by scrolling the well-known Nintendo logo from top to bottom:

```
CE ED 66 66 CC 0D 00 0B 03 73 00 83 00 0C 00 0D
00 08 11 1F 88 89 00 0E DC CC 6E E6 DD DD D9 99
BB BB 67 63 6E 0E EC CC DD DC 99 9F BB B9 33 3E
```

After that it compares them with the ones stored in the BootROM and if they are different the console hangs.

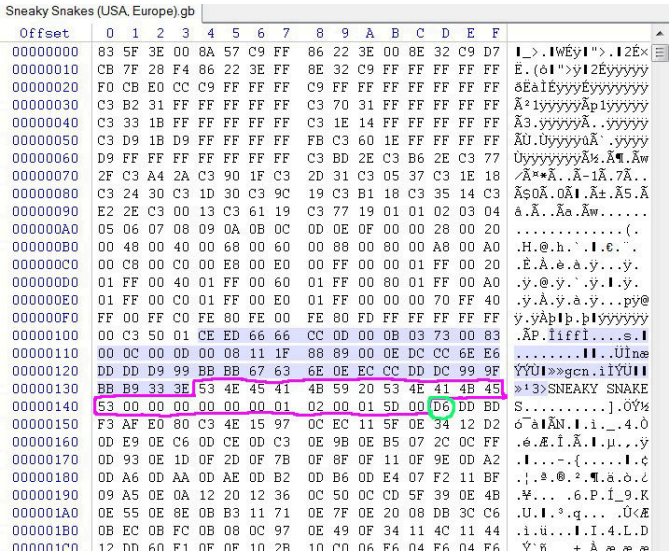


Reading the logo directly from the cartridge explains why, if you turn on a Game Boy without inserting any game, this black rectangle appears (the bytes of the logo are all at FF):

For a full description of how logo bytes work I refer you to this wonderful article:
https://catskull.net/gameboy-boot-screen-logo.html

3 - a second check after the logo check is to sum the values from offset 0x0134 to 0x014C and then add the value present in 0x014D: if the sum is zero the program continues otherwise the console hangs.

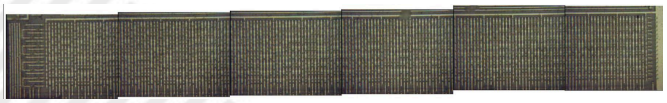
Here is a screenshot of a random GB ROM where you can check the offsets mentioned:



This protection was necessary to prevent other developers from producing software not licensed by Nintendo. As a side note it must be said that SEGA later tried the same approach with the Genesis, but its lawsuit against Accolade did not go as planned:
https://en.wikipedia.org/wiki/Sega_v._Accolade.

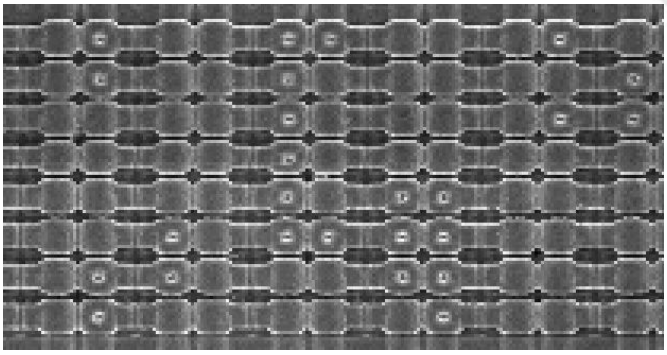
HOW TO BYPASS THE PROTECTION

The content of this mini-operating system remained a mystery until 2003 when the coder Neviksti: http://dot-matrix-game.blogspot.se/2014/01/boot-roms.html performed his own decapping (i.e. a physical "uncovering" of the chip) and read "with the naked eye" its contents, bit by bit (256x8 = 2048 total bits), after photographing them through a microscope; here they are below in all their micro-splendor:

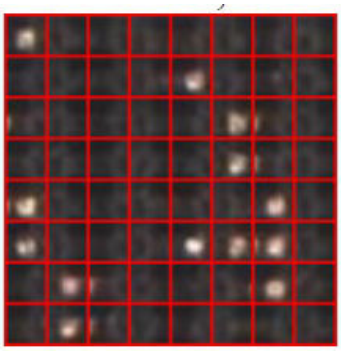


Each "dot" represents a 1-bit, the absence of a dot represents a 0-bit. The dots are nothing more than the electrical charges stored in the memory cells.

Here is an even more zoomed-in example obtained from the decap of another chip seen under an electron microscope (nothing to do with consoles):



Here is an example of how electrical charges are manually read:



```
10000000
00001000
00000100
00000100
10000010
10001110
01000010
01000000
```

Really amazing!

Once converted to bytes and then to ASM, the (commented) code of the BootROM appears to be as follows:

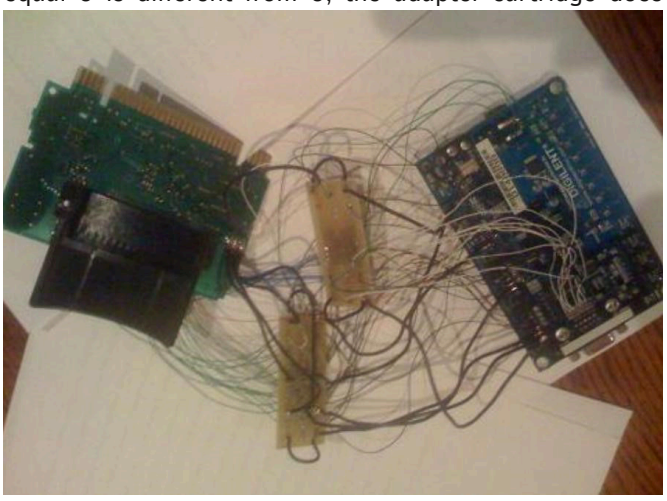
```
LD SP,$fffe ; $0000 Setup Stack

XOR A ; $0003 Zero the memory from $8000-$9FFF (VRAM)
LD HL,$9fff ; $0004
Addr_0007:
LD (HL-),A ; $0007
BIT 7,H ; $0008
JR NZ, Addr_0007 ; $000a

LD HL,$ff26 ; $000c Setup Audio
LD C,$11 ; $000f
LD A,$80 ; $0011
LD (HL-),A ; $0013
LD ($FF00+C),A ; $0014
INC C ; $0015
LD A,$f3 ; $0016
LD ($FF00+C),A ; $0018
LD (HL-),A ; $0019
LD A,$77 ; $001a
LD (HL),A ; $001c

LD A,$fc ; $001d Setup BG palette
LD ($FF00+$47),A ; $001f
```

```
LD DE,$0104 ; $0021 Convert and load logo data from cart into Video RAM
LD HL,$8010 ; $0024
Addr_0027:
LD A,(DE) ; $0027
CALL $0095 ; $0028
CALL $0096 ; $002b
INC DE ; $002e
LD A,E ; $002f
CP $34 ; $0030
JR NZ, Addr_0027 ; $0032

LD DE,$00d8 ; $0034 Load 8 additional bytes into Video RAM
LD B,$08 ; $0037
Addr_0039:
LD A,(DE) ; $0039
INC DE ; $003a
LD (HL+),A ; $003b
INC HL ; $003c
DEC B ; $003d
JR NZ, Addr_0039 ; $003e

LD A,$19 ; $0040 Setup background tilemap
LD ($9910),A ; $0042
LD HL,$992f ; $0045
Addr_0048:
LD C,$0c ; $0048
Addr_004A:
DEC A ; $004a
JR Z, Addr_0055 ; $004b
LD (HL-),A ; $004d
DEC C ; $004e
JR NZ, Addr_004A ; $004f
LD L,$0f ; $0051
JR Addr_0048 ; $0053

; === Scroll logo on screen, and play logo sound===

Addr_0055:
LD H,A ; $0055 Initialize scroll count, H=0
LD A,$64 ; $0056
LD D,A ; $0058 set loop count, D=$64
LD ($FF00+$42),A ; $0059 Set vertical scroll register
LD A,$91 ; $005b
LD ($FF00+$40),A ; $005d Turn on LCD, showing Background
INC B ; $005f Set B=1
Addr_0060:
LD E,$02 ; $0060
Addr_0062:
LD C,$0c ; $0062
Addr_0064:
LD A,($FF00+$44) ; $0064 wait for screen frame
CP $90 ; $0066
JR NZ, Addr_0064 ; $0068
DEC C ; $006a
JR NZ, Addr_0064 ; $006b
DEC E ; $006d
JR NZ, Addr_0062 ; $006e

LD C,$13 ; $0070
INC H ; $0072 increment scroll count
LD A,H ; $0073
LD E,$83 ; $0074
CP $62 ; $0076 $62 counts in, play sound #1
JR Z, Addr_0080 ; $0078
LD E,$c1 ; $007a
CP $64 ; $007c
JR NZ, Addr_0086 ; $007e $64 counts in, play sound #2
Addr_0080:
LD A,E ; $0080 play sound
LD ($FF00+C),A ; $0081
INC C ; $0082
LD A,$87 ; $0083
LD ($FF00+C),A ; $0085
Addr_0086:
LD A,($FF00+$42) ; $0086
SUB B ; $0088
LD ($FF00+$42),A ; $0089 scroll logo up if B=1
DEC D ; $008b
JR NZ, Addr_0060 ; $008c

DEC B ; $008e set B=0 first time
JR NZ, Addr_00E0 ; $008f ... next time, cause jump to "Nintendo Logo check"

LD D,$20 ; $0091 use scrolling loop to pause
JR Addr_0060 ; $0093

; ==== Graphic routine ====

LD C,A ; $0095 "Double up" all the bits of
```

```
the graphics data
LD B,$04 ; $0096 and store in Video RAM
Addr_0098:
PUSH BC ; $0098
RL C ; $0099
RLA ; $009b
POP BC ; $009c
RL C ; $009d
RLA ; $009f
DEC B ; $00a0
JR NZ, Addr_0098 ; $00a1
LD (HL+),A ; $00a3
INC HL ; $00a4
LD (HL+),A ; $00a5
INC HL ; $00a6
RET ; $00a7

Addr_00A8:
;Nintendo Logo
.DB         $CE,$ED,$66,$66,$CC,$0D,$00,$0B,
$03,$73,$00,$83,$00,$0C,$00,$0D
.DB $00,$08,$11,$1F,$88,$89,$00,$0E,$DC,$CC,
$6E,$E6,$DD,$DD,$D9,$99
.DB $BB,$BB,$67,$63,$6E,$0E,$EC,$CC,$DD,$DC,
$99,$9F,$BB,$B9,$33,$3E

Addr_00D8:
;More video data
.DB $3C,$42,$B9,$A5,$B9,$A5,$42,$3C

; ===== Nintendo logo comparison routine =====

Addr_00E0:
LD HL,$0104 ; $00e0 ; point HL to Nintendo
logo in cart
LD DE,$00a8 ; $00e3 ; point DE to Nintendo
logo in DMG rom

Addr_00E6:
LD A,(DE) ; $00e6
INC DE ; $00e7
CP (HL) ; $00e8 ;compare logo data in cart
to DMG rom
JR NZ,$fe ; $00e9 ;if not a match, lock up here
INC HL ; $00eb
LD A,L ; $00ec
CP $34 ; $00ed ;do this for $30 bytes
JR NZ, Addr_00E6 ; $00ef

LD B,$19 ; $00f1
LD A,B ; $00f3
Addr_00F4:
ADD (HL) ; $00f4
INC HL ; $00f5
DEC B ; $00f6
JR NZ, Addr_00F4 ; $00f7
ADD (HL) ; $00f9
JR NZ,$fe ; $00fa ; if $19 + bytes from
$0134-$014D don't add to $00
; ... lock up

LD A,$01 ; $00fc
LD ($FF00+$50),A ; $00fe ;turn off DMG rom
```

In 2009 another reverser, Costis Sideris: https://www.its.caltech.edu/~costis/sgb_hack/, managed to dump the Super Game Boy BootROM using, through an FPGA, a hardware glitch that disabled the last lock out instruction of the memory where the copy of the BootROM was saved, thanks to the sending of overclocked pulses, thus managing to dump the code from offset 0x0000 to offset 0x00FF with specific code inserted from offset 0x0100 onwards. Here is a picture of the "simple" system used:

By examining the code of this BootROM it was discovered that if the second checksum of the summation that must equal 0 is different from 0, the adapter-cartridge does



not disable the game cartridge, but the Super Nintendo does.

Always Costis Sideris and always in 2009 managed to dump also the BootROM of the Game Boy Color, using a combination of glitches this time both clock and power supply, through an FPGA. In this dump he realized that

the check of the logo occurred only for the first half of the bytes and not all 48. Together with it he managed to dump the content of a second ROM of 1792 bytes present in this console.

Dr. Decapitator (aka Christopher Tarnovsky: https://en.wikipedia.org/wiki/Christopher_Tarnovsky) tried a partial decapping, identifying even 3 ROMs (256, 1792 and 512 bytes) but of the 512 bytes ROM it was not possible to obtain any dump (it is possible that it contains the CPU microcode or code related to the LCD).

These are therefore secrets violated after more than a decade. Moreover, in 2012 the patent (https://www.google.com/patents/US5134391) that describes the functions of the BootROM expired, confirming what was previously discovered by the reverse engineers.

Once they discovered how the BootROM works, some developers were ingenious to make the Game Boy read a custom logo from the cartridge at the usual offset (the first stage that allows the display of the logo), but then check a second logo (the official one), stored at another offset of the cartridge.

The results are similar to the following:
The next article will take us to discover the GameBoy Advance protections!

Useful links

Custom logos
At this address you can find a nice description of the various types of custom logos produced by some of the developers:
http://fuji.drillspirits.net/?post=87

BootROM
More information about BootROMs, including one found in a prototype cartridge of Pokemon Stadium for Nintendo 64, missing in the official release of the same:
http://gbdev.gg8.se/wiki/articles/Gameboy_Bootstrap_ROM

GBC BootROM
More information about GBC's BootROM can be found at:
https://tcrf.net/Game_Boy_Color_Bootstrap_ROM

ATTENTION: LIABILITY DISCLAIMER

The information contained in this article is for informational purposes only. This documentation is not guaranteed to be error-free. If this information is used to modify your hardware, it is your responsibility to take all necessary emergency, backup, redundancy, and other measures to ensure safe use. RetroMagazine World assumes no responsibility for any damage caused by the use of the information in this article.

ZX99 and other unofficial add-ons for Sinclair ZX81

by Alberto Apostolo

In the 1980s, several British companies offered expansion boards to increase the versatility of the ZX81.

Most boards either augmented the RAM or were additional circuits for sound effects or joystick interfaces [SU83].

Others allowed you to connect professional printers, save data and programs on floppy disks.

Only a fraction of all the equipment sold at the time is described in the article.

ZX99

ZX99 (Fig.1) was the name of a control system produced in 1982 by Data-Assette of London and sold for £62.90 (including VAT) and then for £49.95 (Fig.2).

The system could connect up to four tape recorders, plus it had an RS232 interface to connect a printer [Ada82].

The routines to manage the recorders were stored in a special ROM allocated in the 8K and 16K section because the ZX99 system considered the recorders as INPUT units or OUTPUT units (two recorders were the minimum allowed). Two other cables connected the card to the EAR and MIC outputs of the Sinclair ZX81 (Fig.3).

I/O operations required machine language routines to be run through the Sinclair BASIC USR command. The result of such operations was obtained with the command LET L = USR 1234 (L = zero indicated that all had gone well). By using BASIC commands and l.m. routines, byte sequences could also be saved. The RS232 interface could operate between 110 and 9600 Baud (Baud = bits per second).

Fig.1

Directions to download the manual are in [DA99].

Finally, Data-Assette proposed other devices [DA98]:

1) ZX21 Metered Loading Interface at 19.95 to measure signal quality,

2) ZX22 Jigsaw Tape Load Intreface for £39.95 (with program loading indicator LED),

3) ZX98 Parallel and serial interface at 99.95 GBP to transform the ZX81 into a video-terminal and take advantage of telematic services.

Memopak

It was card containing a memory expansion manufactured by Memotech of Oxford (Fig. 4).

A kind of manual (with demonstration BASIC programs) can be found in [MEM93].

Forth for ZX81

David Husband's Skywave (Bournemouth) produced a RAM expansion and ROM containing the FORTH language in 1983 (Fig.5). The reference to download the manual can be found in [FORTH20].

Floppydisc Interface for ZX81

The F.I.Z. (Floppydisc Interface for the ZX81, Fig.6,7,8) manufactured in

Fig.2 ("ZX Computings" Oct-Nov. 1982, p.2, www.yumpu.com).

Fig.3

1982 by Macronics (Solihull, West Midlands) provided in the same unit a 5.25 inch floppy disk drive, the management software in 2K ROM [FIZ21][FIZ82].

Each disk had 43K of space with a loading speed of 8K in 22 seconds and was divided into 34 tracks of 1280 bytes (34 x 1280 = 43520).

The management routines contained in the 2K ROM could be called from BASIC programs.

The unit also provided power for the entire system (including the Sinclair ZX81). There were also plans to sell other interfaces (including one to create a local area network) and a printer.

F.I.Z. was sold for 303.03 GBP (VAT included). The motherboard alone cost £79.95 (plus £1.50 for shipping) and was compatible with the Memopak 64 expansion [DA98] but there is no confirmation that it actually worked with that expansion.

MEMOTECH Explores the Excellence of your ZX81
THIS MONTH- 2 NEW PAKS!
MEMOPAK 32K
MEMOPAK 16K
MEMOPAK 64K
MEMOPAK 32K RAM
Operates in tandem with Sinclair 16K Ram to give a full 48K!
CENTRONICS 1/F
Memotech's Memopak Range
All five of the currently available Memopaks are housed in elegant black anodised aluminium cases, and are styled to fit wobble-free onto the back of the ZX81, allowing more add-ons (from Memotech or Sinclair) to be connected.
£68.70 plus VAT
MEMOPAK 64K MEMORY EXTENSION
The 64K Memopak extends the memory of the ZX81 by 56K, and with the ZX81 gives 64K, which is neither switched nor paged and is directly addressable. The unit is user transparent and accepts commands such as 10 DIM A(9000).
Breakdown of memory areas...0-8K-Sinclair ROM. 8-16K-This area can be used to hold machine code for communication between programmes or peripherals. 16-64K-A straight 48K for normal Basic use.
£43.45/£26.00 plus VAT
MEMOPAK 32K and 16K MEMORY EXTENSIONS
These two packs extend and complete the Memotech RAM range (for the time being!) A notable feature of the 32K pack is that it will run in tandem with the Sinclair 16K memory extension to give 48K RAM total.
£52.00 plus VAT
MEMOPAK HIGH RES GRAPHICS PACK
HRG Main Features — • Fully programmable Hi-Res (192 × 248 pixels) • Video page is both memory and bit mapped and can be located anywhere in RAM. • Number of Video pages is limited only by RAM size (each takes about 6.5K RAM) • Instant inverse video on/off gives flashing characters • Video pages can be superimposed • Video page access is similar to Basic plot/unplot commands • Contains 2K EPROM monitor with full range of graphics subroutines controlled by machine code or USR function
£34.70 plus VAT
MEMOPAK CENTRONICS TYPE PARALLEL PRINTER INTERFACE
Main Features — • Interfaces ZX81 and parallel printers of the Centronics type • Enables use of a range of dot matrix and daisy wheel printers with ZX81 • Compatible with ZX81 Basic, prints from LLIST, LPRINT and COPY • Contains firmware to convert ZX81 characters to ASCII code • Gives lower-case characters from ZX81 inverse character set
Coming Soon...
MEMOPAK RAM
HI-RES GRAPHICS
CENTRONICS I/F
ZX81
A complete range of ZX81 plug-in peripherals
Digitising Tablet
RS232 Interface
We regret we are as yet unable to accept orders or enquiries concerning the above products, but we'll let you know as soon as they become available.
Please make cheques payable to MEMOTECH Ltd.
Please Debit my Access/Barclaycard account number
Please delete whichever does not apply
Please send me
64K RAM £68.70 + £10.30 VAT £79.00
32K RAM £43.45 + £6.52 VAT £49.95
16K RAM £26.00 + £3.90 VAT £29.90
HRG £52.00 + £7.80 VAT £59.80
CENTRONICS I/F £34.70 + £5.20 VAT £39.90
Packaging & Postage £2.00 per unit
TOTAL ENC
SIGNATURE
DATE
NAME
ADDRESS
TELEPHONE
We want to be sure you are satisfied with your Memopak – so we offer a 14-day money back Guarantee on all our products.
Memotech Limited, 3 Collins Street, Oxford OX4 1XL, England Tel: Oxford (0865) 722102 Telex: 837220 Orchid G
ZX COMPUTING OCT/NOV 1982
11

Fig.4 ("ZX Computings" Oct-Nov.1982, p.11, www.yumpu.com).

"Run more than ten tasks on a ZX81-FORTH ROM?"
Sure! More than 10 tasks simultaneously and, in some cases, up to 300 times faster! That's what replacing the basic ROM with the new FORTH does for the ZX81 – and more!
The brains behind the breakthrough belong to David Husband, and he's building Skywave Software on the strength of it. Already orders are flooding in and it's easy to see why.
The ZX81-FORTH ROM gives you a totally new system. In addition to multi-tasking and split screen window capability, you can also edit a program while three or four others are executing, schedule tasks to run from 50 times a second to once a year, and with a further modification switch between FORTH and BASIC whenever you like.
The ZX81-FORTH ROM gives you a normal keyboard with a 64 character buffer and repeat, it supports the 16k, 32k, 64k RAM packs, it is fig-FORTH compatible and it supports the ZX printer.
The price, too, is almost unbelievable. As a "fit it yourself Eprom", complete with manual, it's just £25+VAT.
Add £2 p&p UK (£5 Europe, £10 outside Europe) and send your order to the address below.
Skywave
SOFTWARE
David Husband
73 Curzon Road, Bournemouth, BH1 4PW, ENGLAND.
Tel: (0202) 302385.
International +44 202 302385.

Fig.5 [SP84]

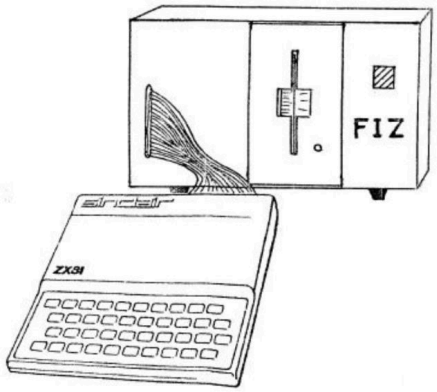


Fig.6

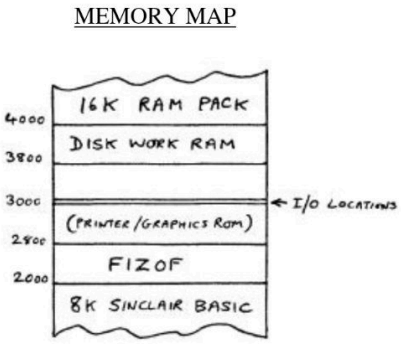


Fig.7

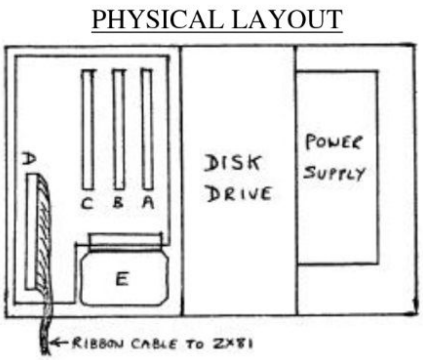


Fig.8

Bibliography

[Ada82] S.Adams, "Control system boosts power", Sinclair User n.5, Aug. 1982, https://www.sinclairuser.com/005/zx99.htm

[DA98] Data-assette (comm.adv.), Sync n.3, May-Jun 1983, pag.116, https://archive.org/details/syncmagazine-v3_n3

[DA99] Data-assette, "ZX99 User Manual 2K Rom", Interact Software Limited, 1982, https://archive.org/details/zx99-user-manual

[FIZ21] FIZ , retrieved (2021/08/14) from https://spectrumcomputing.co.uk/entry/1000151/Hardware/FIZ

[FIZ82] Sinclair User n.2, May 1982, (2020/08/19) retrieved from https://www.sinclairuser.com/002/hardwre.htm

[FOR20] Skywave Software, "Sinclair ZX81-FORTH Rom with Multi-Tasking" (2010/08/19) retrieved from http://zx81.ordi5.free.fr/dominique/download/zx81-forth-h4th.pdf

[MEM93] Fanzine ZXirQLiveAlive n.4, 1993, pagg.7-8, https://archive.org/details/ZXirQLiveAlive_Issue_0304/mode/2up

[SP84] Sinclair Programs n.18, Apr 1984, https://datassette.nyc3.cdn.digitaloceanspaces.com/revistas/sinclairprograms18-apr84.pdf

[SU83] Sinclair User n.12, Mar 1983, http://www.retro8bitcomputers.co.uk/Content/downloads/magazines/sinclairuser/SinclairUser_012_Mar_1983.pdf

Analogue Pocket
the revolutionary multi-videogame handheld!

by Franco Bressan

Analogue is a company that has made a name within the retro community not only for the high price of its devices but also for the perfect reproduction of the games through FPGAs trying to reproduce the original consoles without using emulators.

The Analogue Pocket maintains this concept, working exactly as the Nintendo's portable consoles and, through separately sold modules, Game Gear, Neo Geo Pocket, Atari Lynx and TurboGrafx-16 without ever emulating anything.

But if you want to play GameBoy Color/Advance games, why would I choose the Analogue pocket over the Nintendo handheld?
For the main element of the Pocket: the screen.
It mounts a 3 ½ inch LCD screen at a resolution of 1600x1440, exactly 10 times the resolution of the original 8-bit by Nintendo.
This allows you to perfectly scale our favorite titles in high resolution, with faithful color reproduction and all in a high quality backlit screen. Other comforts that the Pocket introduces is the ability to put the game on standby and resume directly from where we stopped, in future updates will also be included the ability to create save states, to recreate the characteristics of the display of the original hardware and for those who dabble in the development of Game Boy games, Analogue allows you to develop new titles with GB Studio that has added the Pocket among the compatible devices. It also has an integrated DAW for the production of soundtracks.

The pocket also works with cartridge adapters from other systems such as Game Gear, Neo Geo Pocket, Atari Lynx and many others. A project that does not emulate but is in FPGA.

Among the features the one that amazes the most is the ability to make music. The pocket has a digital audio workstation called Nanoloop. It's a synthesizer and sequencer designed for music creation and live performances. A joy for chiptune lovers.

You can also connect it to your PC, Mac or other through the Midi cables that you will find for sale separately.

To finish, you can connect the console to your television with an HDMI adapter. You can connect 8BitDo Bluetooth controls wirelessly or controlled via wired USB.

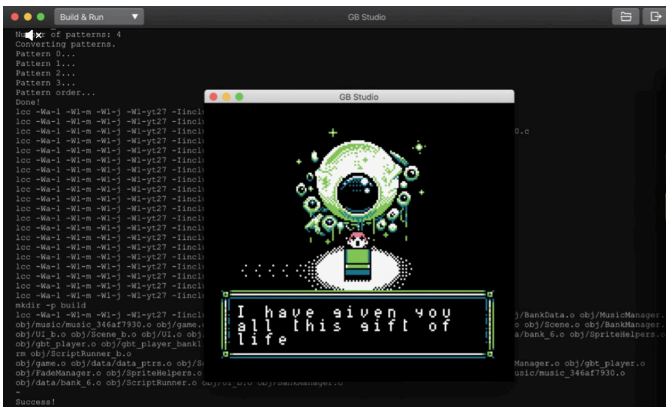
TECHNICAL SPECIFICATIONS:

Console

- Compatible with Game Boy, Game Boy Color, and Game Boy Advance game cartridges
- 3.5" LCD. 1600×1440 resolution. 615ppi.
- 360° display rotation (tate mode)
- Rechargeable 4300 mAh lithium-ion battery
- 6-10 hours of play and over 10 hours of sleep 9
- All buttons mappable
- Stereo speakers
- micro SD card slot
- USB-C Charging
- Original style connecting door
- 3.5mm headphone output

Dock

- 1080p HDMI output
- Bluetooth and 2.4g support for wireless controller
- 2 USB inputs for wired controllers
- Support up to 4 players (4p Bluetooth, 2p 2.4g, 2p USB wired)
- DAC compatible
- Power Dock turned on by controller



For more information and orders please visit the official website: https://www.analogue.co/pocket

Fairchild Channel F

di Leonardo Miliani

In the last article we examined the genesis of home gaming devices going to cover all the secrets of what is absolutely the first home console put on the market, the Odyssey of Magnavox. Following that thread, in this article we're going to get to know a little star, in her own way. The protagonist of our story boasts several records: it is reputed to be the first console of the second generation, it is the first console to use a microprocessor, and it is the first cartridge-based console with interchangeable ROM memories to load games. We are talking about the Fairchild Channel F (fig. 1).

RAVEN

To fully understand the origins of the Channel F, we need to start with the context in which the idea that led to its creation was born, and to do this we need to go back several decades, we need to go back to the USA in the 1960s. At this time in Stamford, Connecticut, there is the research and development department of AMF (American Machine & Foundry), a multi-faceted company that manufactures everything from garden furniture to nuclear reactors (!), and here scoreboards for bowling alleys are produced. At some point the employees working there get the news that the facility is moving to North Carolina. Not everyone is excited about moving 500 miles to another state and some decide to leave the company. Among them was an engineer named Norman Alpert who, together with other former AMF employees, founded Alpex Computer. Initially the company was dedicated to the production of electronic cash registers but business slowly took a turn for the worse and Alpex Computer, in the early '70s, found itself in full crisis. Another ex-employee of AMF, Wallace Kirschner, comes up with the idea that a possible way to avoid bankruptcy is to enter the emerging market of electronic video games. This idea comes to fruition when he sees how the first home video game console, the Magnavox Odyssey, is enjoying good success, and how Atari's Pong arcade is literally blowing up in bars and pubs.

Fig. 1: Fairchild Video Entertainment System, after named Channel F (picture di Evan Amos - source: Wikimedia Commons)

However, these games have a defect: they cannot be changed. Pong, and the arcades that follow it, have a motherboard composed of dozens of logic circuits that create the video game on the screen and manage the interaction with the player, but if the manufacturer wants to change the game, he is forced to redesign the board from scratch. Even the Odyssey, although "programmable" through external boards, is actually a console with passive components and the game boards work like switches and switches to change some electrical lines and so vary the logic of operation of the console, proposing variants of the only game scheme integrated in it. But things are about to change, because in those years Intel presented in 1971 the 4004, the first monolithic microprocessor in history, followed the following year by the 8008. This is a small revolution because finally you can concentrate a lot of transistors in a single small container creating a component that can load and execute instructions read from an external memory whose content can be modified to change the program executed. Kirschner understands that this is the right way and proposes his idea to Alpert who immediately endorses the project of a video game but also understands that it is a race against other competitors who are throwing themselves into that field and that this race can not win with the few employees of his company. So he decides to hire Lawrence Haskel, an

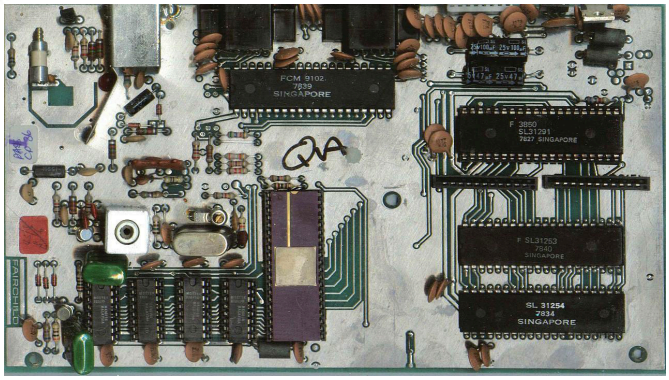


Fig. 2 - The motherboard of the Channel F System II. Note the few integrated circuits present (photo: Incog88 - source: Wikimedia Commons)

engineer he has worked with before in his days at AMF. Haskel also has an advantage, he's a video game enthusiast himself so it's a dream come true for him to be able to work on such a project. So, at the beginning of 1974, Kirschner and Haskel started working on a project they called "Remote Access Video Entertainment", or RAVEN: the first one worked on the hardware part while the second one programmed the games. As the "electronic brain" they chose the 8008, the first 8-bit processor ever produced, to which they added a circuit capable of generating an image of 128x64 pixels on a common TV screen. Such a low resolution is dictated by the fact that at the time memories are really expensive and to manage an image of 8,192 total pixels (128x64=8192) you need 8 kilobits of memory (8,192 bits = 1 Kbyte) where each bit represents a pixel (on or off) on the screen. During development, Kirschner adopted the more powerful 8080 processor as soon as it was released by Intel.

Haskel, for his part, creates the first game inspired by Atari's Pong but making it richer and more similar to field hockey, a sport that was very popular in the cold winters of northeastern America. He realises a field of play by outlining it with borders and area lines, he adds rackets that not only can be moved vertically but also horizontally, thus being able to be moved throughout the playing area. It also adds the ability to rotate them so that the ball bounce varies depending on the position of the racket. It also adds "hockey goalie leg pads" (if you have ever seen a field hockey game, you have those pads tied to the goalkeepers legs in mind), barriers that can only be moved vertically to defend your goal. After the "Hockey" program the "Tic-Tac-Toe", the "Shooting Gallery" (where the player must rotate a "gun" and shoot at a moving target) and "Doodle", a primitive art program where the player must draw or erase lines on the screen. To allow all the movements of the elements of his games, Kirschner made two gamepads with numerous buttons, each of which corresponds to a command such as go up, down, rotate, shoot, etc..

To solve the issue of interchangeable games, a very simple solution is adopted. They buy from RadioShack some rectangular plastic containers with dimensions very similar to those of the common cassette tapes of the time, and they put inside them a small card with an EPROM mounted on it: each cartridge contains a different one, for each of the games made, and it is inserted in a connector of the motherboard of the console. The choice of EPROMs comes to them from the development board that Intel distributes for the 8080, a board containing precisely this type of memory that can be deleted and reprogrammed to correct program errors or to load new versions of the same.

From RAVEN to Fairchild Video Entertainment System

The prototype is presented to Alpert, who finds it very interesting, but also realizes that his small company would never be able to find the resources to market it on its own. He therefore decided to find a commercial partner, and some TV manufacturers were the first to be contacted, thinking that a product of this kind to be sold in combination with their TV sets could arouse their interest. But things don't go as hoped because, despite several demonstrations, no one wants to invest in it. Alpert decided to contact Fairchild Semiconductor, a large integrated circuit

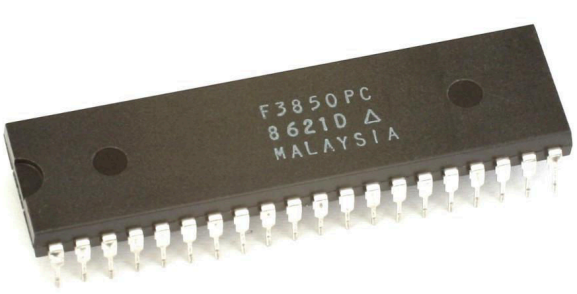


Fig. 3 - The Fairchild F8 CPU (photo: Konstantin Lanzet - source: Wikimedia Commons)

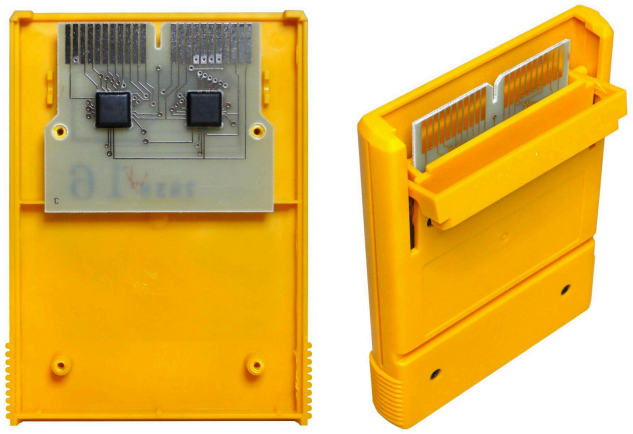


Fig. 4 - The inside of a Videocart and the contact protection flap (photo: Schnurrikowski- source: Wikimedia Commons)

manufacturer from whom he regularly purchased components for his products. An engineer is sent to look at the RAVEN, which is considered interesting: the report also intrigues the vice-president of the consumer department, who in turn talks to the CEO Wilf Corrigan of Fairchild Camera and Instrument, the parent company. Corrigan sends Gene Landrum and Jerry Lawson to Alpex Computer to thoroughly evaluate the prototype, which is evaluated as commercially interesting. Fairchild collaborates with Alpert to finalise the console and transform the RAVEN into a real videogames console, supporting Kirschner and Haskel with Lawson. First of all the F8 CPU from Fairchild was adopted (fig. 3), essential requirement to establish the collaboration (on the other hand, a manufacturer of microprocessors would never put on the market a product based on a CPU of a direct competitor ...), and then Lawson works to simplify the control system based on the keyboard used on the RAVEN, entrusting the project to Nicholas Talesfore, who made the iconic vertical joystick with movable head. Meanwhile the project is renamed "STRATOS".

After these changes, STRATOS is approved for marketing and we start to design the final version that must be smaller than the RAVEN, such as to allow it to be placed on top of a common TV. Another point to be fixed are the cartridges. Putting aside the patched case used by Kirschner and Haskel, Talesfore entrusts the task to Ron Smith, a mechanical engineer with whom he has worked in the past and who has already made an interchangeable memory system for a National Semiconductor calculator, even though it had been cancelled by the company before coming to market. The cartridges are designed to be shock-resistant and with a flap covering the comb with the connection pins that opens automatically once inserted into the appropriate slot in the console: once in their place, a locking mechanism keeps them anchored to prevent them from being inadvertently pulled out. To make them familiar to the console's purchasers, they are made to be about the size of the Stereo8 cassettes then in use for music, and made in a flashy yellow colour so that they are easily identifiable (fig. 4). The console case is made of wood and dark plexiglass with buttons for power, reset and selection of integrated games as well as the various modes on the front panel. Particularly noteworthy is the presence of a pause button to temporarily stop the running game. On the back of the top side there are 2 slots to store the joysticks when not in use: these are not separable from the console but permanently fixed with non-unpluggable cables.

The console in its final form made its first official debut at the Summer Consumer Electronics Show in Chicago in June 1976 as the Video Entertainment System, or VES. During the event, however, it does not attract the attention of press and visitors because what Fairchild shows is a simple empty shell, with no functioning electronics. After that first appearance it makes a more dignified appearance in a Businessweek article of July 6, 1976 entitled "The Smart Machine Revolution" where the futuristic and countless applications of the emerging microprocessors are shown, appearing alongside other electronic devices. The real commercial debut occurs towards the end of the year: the console is advertised in print and television and put on sale at a price of $169.95. Sales initially go well because the technical leap from the previous Odyssey is remarkable, and the possibility of having new games in addition to those that can be found preinstalled attracts buyers.

Technical Analysis

The console is built around the Fairchild F8 microprocessor.

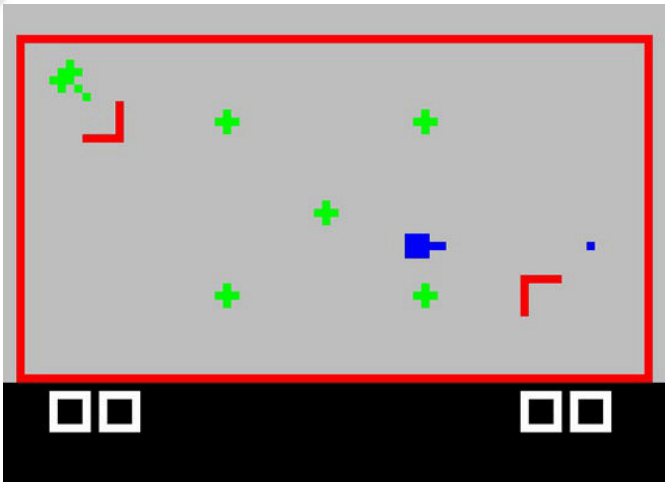


Fig. 5 - The Desert Fox game (source: voxodyssey.com)

Developed in 1971, it is actually a "chipset", that is a set of integrated chips that go side by side to form the processor itself. The configuration used in the first version of the console sees the use of Fairchild 3850 and Fairchild 3851 (fig. 2). The first integrated is the CPU itself, containing the ALU (the logical arithmetic unit), 2 input/output ports and 64 bytes of RAM available in the form of internal registers: in addition to these bytes the CPU has no other dedicated RAM, so the games themselves provide extra memory when needed, such as "Maze" and "Chess". The integrated 3851 is instead called PSU, for "Program Storage Unit", and contains 1 KB of ROM, other 2 input/output ports, a control bus, the management of an interrupt, the stack register, the Program Counter that points in memory the next instruction to execute. In the ROM are stored not only the BIOS to manage the console at power-on and start the games on the cartridge but also the integrated games. Since the philosophy that had led to the realisation of the F8 was to reduce the characteristics of each chip to the bare minimum, the minimum configuration used was precisely that proposed on the console, consisting of CPU 3850 and PSU 3851. In these 2 chips there is everything you need to make a microcomputer: CPU, ALU, RAM, ROM and Program Counter. It is the PSU that retrieves from its internal ROM the instruction to execute and passes it to the CPU via a dedicated bus: for this reason neither the 3850 nor the 3851 have an external address bus. However, the PSU has a 5 lines "command bus" specifically designed to manage particular ROMs designed to work with F8 and provided with their own internal Program Counter that they manage themselves. The PSU therefore sends "commands" to retrieve data from the ROM and the ROMs retrieve the required data, depending on the command received and the value of their Program Counter. The first game cartridges are equipped with 2 chips of these special ROMs of 1 KB each, connected to the console through a 22 pin connector where the lines necessary for data exchange pass. The frequency of the CPU is 1.79 MHz, obtained by halving the clock of the video signal generator of colour (colour burst), of 3.58 MHz.

Compared to RAVEN, the video buffer is increased to 2 KB, thanks to which the console can generate a video image with a maximum resolution of 128x64 pixels and 2 colour bits for each pixel (4 colours from a palette of 8): going too high with the resolution would have meant having to use more memory with a considerable increase in production cost. The VRAM (video RAM) is composed of 4 DRAM 4096 chips each offering 4 Kbit capacity, for a total of 16 Kbit bits, or 2 Kbytes (128x64 = 8,192 pixels = 8,192x2 bits = 16,384 bits = 2,048 bytes). Not all pixels are usable or viewable, however. In fact, cells 125 and 126 control the palette in use on the corresponding row of the screen while the actual image starts from row 4 and column 4 and extends 102 pixels horizontally and 58 pixels vertically: these are therefore the actual dimensions of the image on the screen. Add to this the fact that not all televisions of the time could adjust the width of the image to display the image, so often the outermost portions horizontally were lost, further reducing the screen size

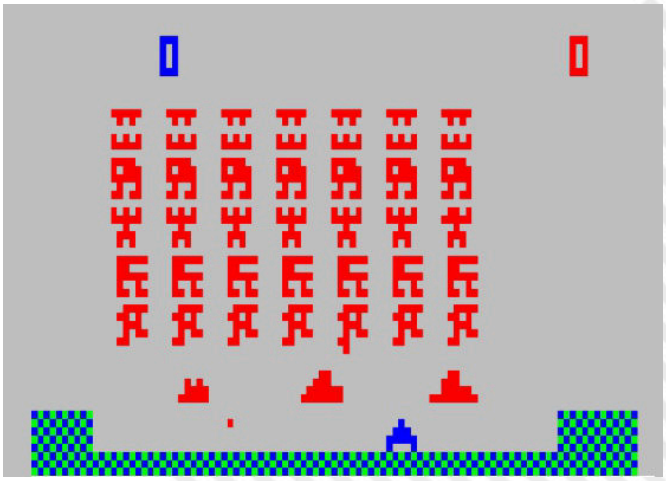


Fig. 6 - Alien Invasion, an obvious Space Invaders clone (source: voxodyssey.com)

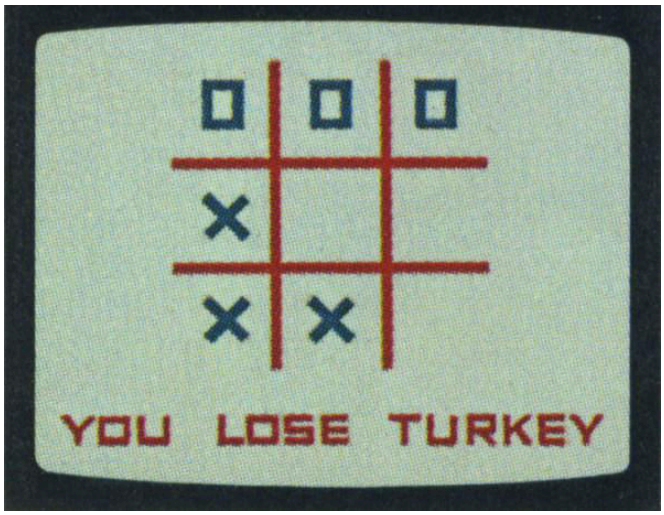


**Fig. 7 - "You lose turkey!"
(source FastCompany.com)**

to 95x58 pixels. The image displays white pixels if you set the background to black, or green, red, and blue pixels if you choose a colored background.

The joysticks are a true electromechanical marvel. The triangular head not only reads the 8 classic horizontal directions but can also be rotated clockwise or counterclockwise, as in Pong-style paddles, and can also be pulled up and pressed down, thus offering a whole range of commands never offered by any other game controller until then seen on the market. The audio is limited to 3 tones with 3 fixed frequencies at 120 Hz, 500 Hz and 1 KHz, reproduced by an internal speaker.

The games available initially are five, namely those initially developed by Haskel: 2 are contained in the ROM of the console itself, "Tennis" and "Hockey", and 3 are published on a cartridge called "Videocart-1": "Tic-Tac-Toe", "Shooting Gallery" and "Doodle". During the conversion of the code from assembly 8080 to F8 Lawrence inserted in "Tic-Tac-Toe" what is considered to be the first joke contained within a video game. At that time, as mentioned, Lawson was leading the RAVEN to STRATOS conversion project, and one of his favourite epithets to call people is "turkey". Lawrence decides to take that epithet and insert it in the game so that it will appear on the screen in case the player loses the game (fig. 7). Curiously enough, Lawson did not realise this immediately, but much later: however, it seems that he took it well, so the joke worked! To these games follow, in the course of the years in which the console remained in production, other titles published exclusively by Fairchild first and then by Zircon, which will take over the rights on the console, for a total of 27 cartridges (some with more than one game). Initially, each cartridge is offered for sale at a price of $ 19.95 each, a not insignificant price if you relate to the initial cost of the console.

Channel F and derivatives

The release of the VES pushed several manufacturers to accelerate the development of their systems to avoid accumulating excessive delay against the console of Fairchild. Among the competitors was the Atari Video Computer System (later became 2600) whose release pushed Fairchild to change the name of its system to avoid confusion among buyers, becoming "Channel F", where the "F" was an abbreviation of "Fun". The release of the Atari VCS also marked the beginning of the decline of Channel F, outclassed by its rival in terms of sales. Atari's VCS was sailing along at an unimaginable pace: it had already sold 500,000 units in its first year of release, and in 1979 alone it sold one million units. On the other hand, until 1979 Channel F had sold a total of 350,000 units... Fairchild slowly lost interest in the video game market and in 1979 sold the rights to the console to a company called Zircon International. The fight with the VCS was unequal. To make the console more attractive and try to make up for some of the gap, Zircon revamped the Channel F's exterior with a more compact and modern design and equipped the console with controllers that could be separated from the body. Audio was improved

Fig. 8 - The Luxor Video Entertainment Computer, version for the Swedish market (photo: liftarn - source: Wikimedia Commons)

by eliminating the internal speaker and mixing it with the RF signal to play through the TV. The new unit was called the Channel F System II. It was also licensed to several manufacturers to expand penetration into other markets, especially Europe (Fig. 8). However, this was not enough to recover market share and the console was permanently removed from the market in 1983, when its games were judged as "antiquated" and the graphics and sound "primitive".

Commercial failure

The basic problem with Channel F was that it was made by a company whose primary goal was not to make games. Channel F was developed by Fairchild Camera and Instrument, a multinational company that went all over the place with several subsidiary companies: the management of the console was entrusted to Exetron, the division responsible for consumer products, which bought the F8 CPUs from Fairchild Semiconductor, the division dedicated to the production of electronic components and integrated circuits. In fact, the parent company saw the console as primarily a vehicle to sell chips to Fairchild Semiconductor. Atari, however, was born as a company devoted exclusively to the production of video games, and every effort was devoted to the realisation of the best possible game. And in fact the titles produced for the 2600, despite the hardware limitations of the machine, outclassed by several orders of magnitude those of the competition, including Channel F: the 2600 offered dynamic, adrenaline-filled titles, full of colours and sound effects as well as action-packed, while Channel F offered "classic" games, slow, monotonous, too "calm" for the hungry gamers of the time who were beginning to explore a newly discovered world. The 2600's game library grew monthly, while Channel F's was producing a few dozen titles in 7 years of life: there could be no comparison. And there wasn't.

Legacy

As mentioned in the beginning, the Channel F boasts several records, including being the first console based on a microprocessor, specifically the Fairchild F8. Having a microprocessor on board allowed the console to be "programmed", i.e., to execute even complex programs: the Odyssey moves the rackets on the screen simply by following the player's commands, but here there is an "electronic brain" that allows to perform calculations independently and give the console a primitive form of artificial intelligence sufficient to offer, for the first time, the possibility to play "against the computer" and no longer only against another human player. "Tic-Tac-Toe", "Baseball" and "Checkers" are examples where the opponent to beat is moved by the CPU.

In truth, interchangeable cartridges were not a novelty because before that time they had already been used to allow changing the software of some calculators made by various manufacturers, including National Semiconductor, Texas Instruments, HP and others, but it was the first time that such a system was adopted for an accessory that would be manipulated by thousands, tens of thousands of people without special care and for who knows how many times a day. The cartridges were a real unknown, a gamble for the designers: no one had data about how long they would "survive" in the hands of children, there were no statistics on which to base predictions. For this reason they were studied and designed to be as robust as possible, and equipped with every device to reduce possible damage such as the locking system when inserted or the door to protect the comb of contacts that opened automatically when they were inserted. But from that idea cartridges have spread so much to be for more than two decades the exclusive distribution system for games of all consoles that came after the Channel F, dictating law until the mid-90s when CDs and DVDs began to replace them as a cheaper and more spacious media. And they never completely disappeared because portable consoles continue to use them to this day.

In short, today Channel F is an ageing lady who nevertheless turned a lot of heads in her youth.

FOCAL language on PDP-8 - part 1

by Francesco Fiorentini

In the last issue, I've shown you how to reproduce the game Planet Raith in GEOS with BeckerBASIC and I told you the story of its ancestor, King of Sumeria, a game written in 1968 on PDP-8 using the FOCAL language. Unfortunately in that article I was not able to deepen the topic, so I would like to remedy this with these pages.

The PDP-8

The PDP-8 is a minicomputer with 12-bit architecture produced by **DEC**, **Digital Equipment Corporation** in the sixties and, as its name suggests, the eighth project in the series DEC PDP, Programmed Data Processor.

The PDP series owes its name to a curious marketing decision, closely related to the company name. Ken Olson and Harlan Anderson founded DEC in 1957 and initially wanted to call it DCC, Digital Computer Corporation, due the intention of building computers. However, the other shareholders insisted on removing the word computer from the company's name and waiting to build them. This was because, at a time when computers were huge, expensive, and required highly specialized personnel, they did not yet believe in this business. For the same reason, when the first models of the PDP-1 series were produced and put on sale, at a price of just over $100,000 while computers at the time still cost millions of dollars, they avoided using the word computer.

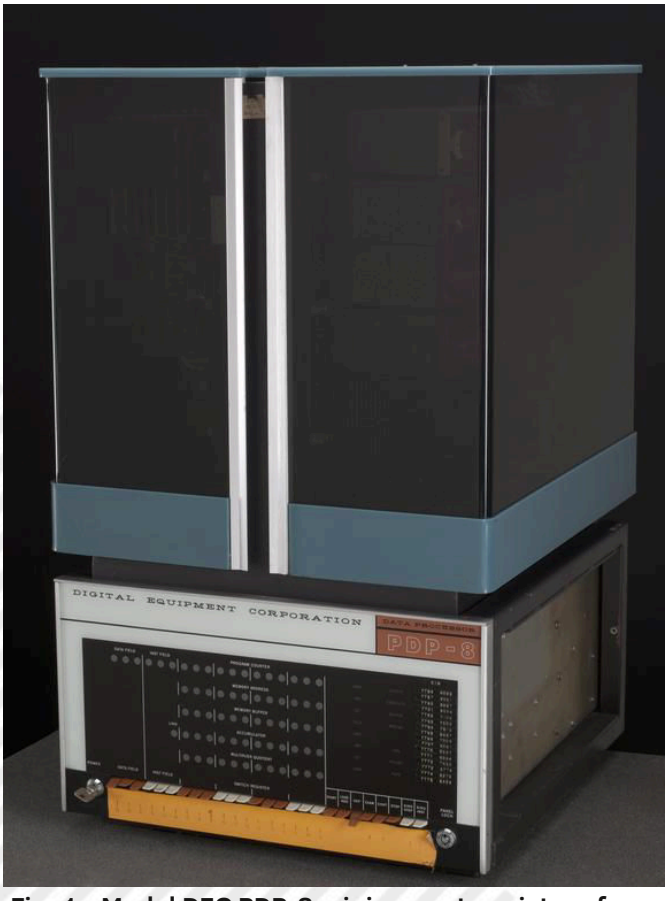


Fig. 1 - Model DEC PDP-8 minicomputer, picture from Science Museum Group Collection

The PDP series obtained excellent results and the PDP-8 had a great sales success, it was in fact purchased by schools, universities and research centers, especially in the United States, recording almost 300,000 units sold. Thanks to this result the PDP-8 has been compared to the Ford T, being the first computer to be mass produced at a cost accessible to (almost) everyone.

Technically, the PDP-8 uses a 12-bit memory address space, so the basic configuration has a main memory of 4,096 twelve-bit words. Instructions have a 3-bit opcode, so there are only eight instructions.

Opcode			Indirect	Page	Address						
11	10	09	08	07	06	05	04	03	02	01	00

There are only three registers visible to the programmer: a 12-bit accumulator (AC), a program counter (PC) and a carryover flag called a link register (L).

Early versions of the PDP-8 did not have an operating system and programs were entered directly into binary machine code using switches on the front panel.

Over the years, operating systems such as OS/8 and COS-310 were developed with an online editor, which allowed the development of command-line compilers for languages such as PAL-III assembly, FORTRAN, BASIC, DIBOL (Digital's Business Oriented Language) and FOCAL.

The FOCAL language

FOCAL, an acronym for FORmula CALculator, is an interpreted programming language developed by DEC for its PDP series of machines. The language made its first appearance in 1968 on the PDP-8 minicomputers

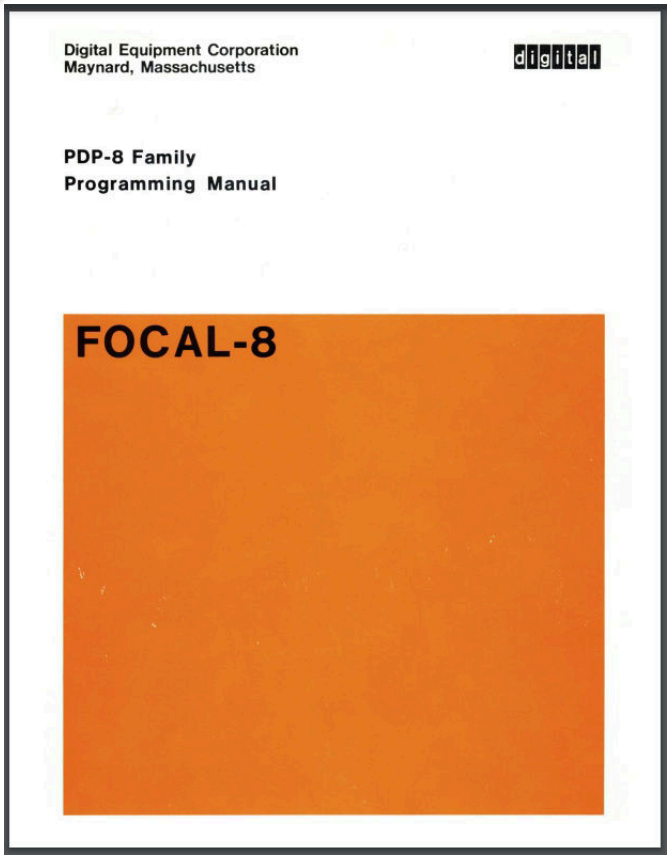


Fig. 2 - FOCAL-8 Programming Manual

and took the name FOCAL-8. A year later an updated version of the interpreter was released and called FOCAL-69. This is the version we will use in our tutorial later on.

The FOCAL language was evidently influenced by another language, JOSS, acronym of JOHNNIAC Open Shop System that had been successfully implemented on the PDP-6 series machines. Like JOSS and later BASIC, of which FOCAL is one of the ancestors, FOCAL on the PDP-8 is a complete programming environment, including editor, interpreter and I/O routines.

The language is composed of imperative commands and mathematical expressions generally in standard notation, making it very easy to learn and handle. As in BASIC, it is also possible to use some commands in immediate mode, without specifying the line number and receiving the result immediately after pressing enter.

A peculiarity of the FOCAL language, or at least of FOCAL-8 in its original configuration, is that it does not handle strings and accepts alphanumeric values as input, for example to assign values to variables, only to transform them into numbers. We will see some examples of this behavior later.

A little practice...

After having introduced you to the PDP-8 and the FOCAL language, I want to show you how to practically put your hands on it and write some simple programs, as if we were in the late sixties, well before most of the machines we usually write about were built.

Thanks to emulators nowadays we can virtually get our hands on almost any man-made computer and, of course, the PDP-8 is no exception.

Among the many emulators present, I have chosen this one realized in python for a couple of reasons. First of all it seems to work well and secondly the source is open, which would eventually allow developers to put their hands on it to produce a fork or to make small changes as I did. Personally I found the image of the front panel too bulky, so I decided to halve the size, modifying the emulator code to handle the new setting (if someone is interested in this change, please write to me).

Equipment and instructions

Python:

- https://www.python.org/downloads/

Pygame library:

- https://www.pygame.org/news

PDP-8 emulator and programs:

- https://github.com/mdoege/PDP-8/

Once you have downloaded and installed python, run this command from the command line to automatically install the pygame library without having to download it previously:
python3 -m pip install -U pygame --user

Download the PDP-8 emulator written in python by mdoege from his github repository: https://github.com/mdoege/PDP-8/

Place yourself inside the folder where you saved the files downloaded from github and, to launch the PDP-8 emulator, run the following command:
python pdp8.py

At this point to load the FOCAL interpreter, use the following

commands:

```
PC 0000 AC 0000 L 0 SW 0000 IE 0
> load focal                    to load the FOCAL module
PC 0200 AC 0000 L 0 SW 0000 IE 0
> r                             to execute the FOCAL module
```

If everything is configured correctly you should see something similar to Fig. 3.

The PDP-8 informs you that the FOCAL 1969 interpreter has been successfully launched on your PDP-8 emulator:

```
CONGRATULATIONS!!!
YOU HAVE SUCCESSFULLY LOADED 'FOCAL,1969'
ON A PDP-8 COMPUTER.
```

And ask you 2 questions:

```
SHALL I RETAIN LOG, EXP, ATN ?:NO
SHALL I RETAIN SINE, COSINE ?:NO
PROCEED.
*
```

The interpreter shows you the options available to the user for maintaining mathematical functions. If these functions will not be needed for the program you intend to implement, you can disable them by answering NO and FOCAL will delete these functions from the core; this will free up space for user programs.

Today this choice makes us smile, but on a PDP-8, where space was definitely a problem, it was a necessary option. But exactly how much space do we gain by removing these functions from memory? We can ascertain this by using the LOCATIONS command:

```
*LOCATIONS
3206
3217
3217
5377
```

Hhmmm, okay, now we probably know even less than before.... What do these numbers mean?

Fortunately, the FOCAL-8 PROGRAMMING MANUAL FOR PDP-8/1 comes to our aid. Running the LOCATIONS command causes FOCAL to print four octal numbers (the PDP-8's core memory runs on base-8 numbers) that represent the following locations within the core:

a. 3206 -> start of text buffer

b. 3217 -> end of text buffer

c. 3217 -> end of variable list

d. 5377 -> limit of available memory

To calculate the space available to the user for his programs we need to subtract c from d.

So d-c -> 5377 - 3217 = 2160.

Be careful though, 2160 is a number in base 8, which translated into base 10 is equal to 1136. So we have about 1100 characters available for our program.

This is because we have chosen to eliminate the mathematical functions, otherwise our space would have been much smaller, see the following table:

Dialogue response	Yes/Yes	No/Yes	No/No
Locations	*L	*L	*L
	3206	3206	3206
	3217	3217	3217
	3217	3217	3217
	4617	5177	5377

Keeping all mathematical functions the space at our disposal would be reduced to about 700 characters, about 900 in the case of maintaining the functions LOG, EXP, ATN and about 1100 in the case of no mathematical function maintained, as we saw previously.

The FOCAL-8 controls

The list of basic commands available to the FOCAL programmer is very limited, nothing like the hundreds of commands available in modern languages and far from even those of the poorest BASIC (see BASIC V2 of Commodore 64).

Here is the list of commands: **TYPE, ASK, WRITE, SET, ERASE, GO, GOTO, DO, IF, RETURN, QUIT, COMMENT, FOR, MODIFY**.

The TYPE command

Type is one of the most used commands in FOCAL. It allows you to calculate and print on the screen the result of an operation or of a variable and also works as an immediate command. Let's try to make an example:

```
*TYPE 23/17
= 1.3530*
```

In this case we have asked to divide 23 by 17 and the result shown is 1.3530. By default, in fact, the FOCAL interpreter is programmed to print on the screen a

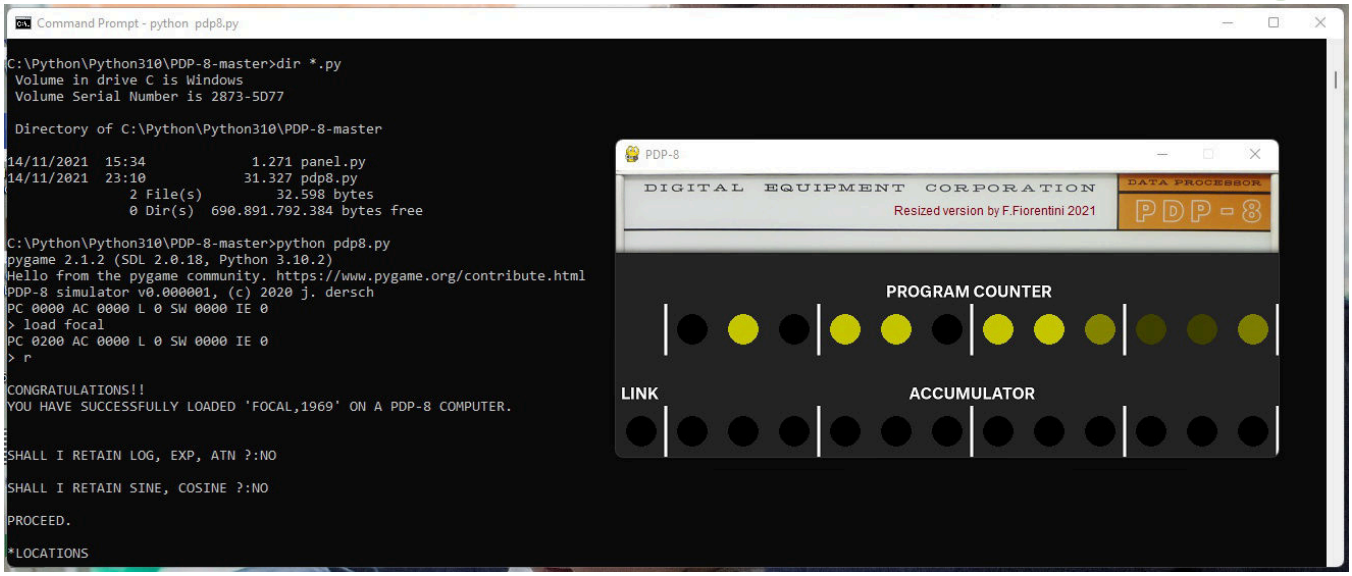

Command Prompt - python pdp8.py
C:\Python\Python310\PDP-8-master>dir *.py
Volume in drive C is Windows
Volume Serial Number is 2873-5D77
Directory of C:\Python\Python310\PDP-8-master
14/11/2021 15:34 1.271 panel.py
14/11/2021 23:10 31.327 pdp8.py
2 File(s) 32.598 bytes
0 Dir(s) 690.891.792.384 bytes free
C:\Python\Python310\PDP-8-master>python pdp8.py
pygame 2.1.2 (SDL 2.0.18, Python 3.10.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
PDP-8 simulator v0.000001, (c) 2020 j. dersch
PC 0000 AC 0000 L 0 SW 0000 IE 0
> load focal
PC 0200 AC 0000 L 0 SW 0000 IE 0
> r
CONGRATULATIONS!!
YOU HAVE SUCCESSFULLY LOADED 'FOCAL,1969' ON A PDP-8 COMPUTER.
SHALL I RETAIN LOG, EXP, ATN ?:NO
SHALL I RETAIN SINE, COSINE ?:NO
PROCEED.
*LOCATIONS
PDP-8
DIGITAL EQUIPMENT CORPORATION
Resized version by F.Fiorentini 2021
DATA PROCESSOR
PDP-8
PROGRAM COUNTER
LINK
ACCUMULATOR

Fig. 3 - The PDP-8 emulator in action with the panel in the foreground

numerical value consisting of 4 integers and 4 decimals (unnecessary zeros are suppressed in favor of a space).

Now try typing the following instruction:

```
*TYPE 23/17, 34+59
= 1.3530= 93.0000*
```

As you can see, we can compute several expressions in one line by separating them with a comma. Another thing you have probably already noticed is that the TYPE command does not force a carriage return after printing the result. To insert a carriage return after each expression and at the end of our calculation, we can use the ! (exclamation mark).

```
*TYPE 23/17,!,34+59,!
= 1.3530
= 93.0000
*
```

The TYPE command also allows us to specify the format in which to display the result. If we want to increase the number of decimals we can use the following notation:
TYPE %x.yz

Where x is the number of total characters in output and yz is the number of decimals

```
*TYPE %9.06.23/17,!,34+59,!
= 1.352940
= 93.000000
*TYPE %10.05.23/17,!
= 1.35294
```

Obviously, priority is always given to significant numbers.

```
*TYPE %9.06,1234+9999
= 11233.0000*
```

Well, for the moment I stop here, I think I have already proposed several interesting ideas to begin to explore this language on your own, of course if you want.

If, on the other hand, you are curious but don't have time or desire to study it on your own, see you in the next issue to continue our adventure with the FOCAL language and to write a first working program.

Useful links

Here two links to as many FOCAL manuals.

FOCAL-8 PROGRAMMING MANUAL FOR PDP-8/1, PDP-S/I, PDP-B/S, PDP-8, LINC- 8, PDP-12, PDP-5:
http://bitsavers.informatik.uni-stuttgart.de/pdf/dec/pdp8/focal/DEC-08-AJAB-D_FOCAL_Programming_Manual_Jan70.pdf

FOCAL Programming Manual for PDP-S PDP-SIS PDP-S/I LAB-S LINC-S:
http://www.bitsavers.org/www.computer.museum.uq.edu.au/pdf/DEC-08-AJAB-D%20PDP-8-I%20FOCAL%20Programming%20Manual.pdf

8-Bit fractals in BASIC sauce - part 3

by David La Monaca

We have reached the third appointment of this column and the amazement in front of what our little 8-bit home computers are able to create with their BASIC languages and some well-designed algorithms remains unchanged. Clearly the Achilles heel of our fun is the slow processing of computing points and lines to be drawn in high resolution.

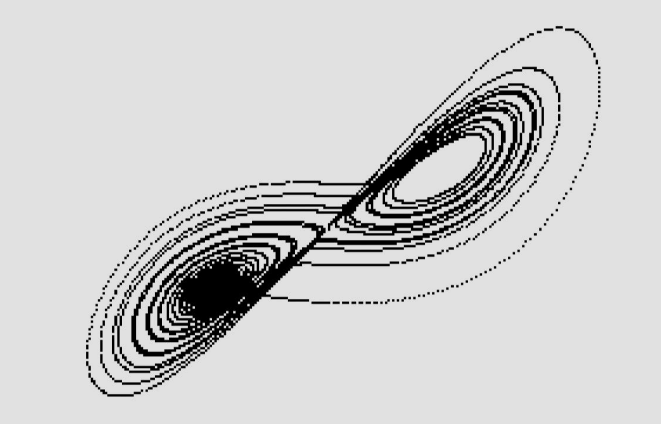
Many of the programs we have published so far are not optimized for maximum speed and efficiency, especially since the objective of our little trip is to have fun seeing how these machines are able to generate the classic fractals and other 2D or 3D geometric figures of general interest.

For those who do not own the original machines and use emulators or FPGA systems, sometimes a "Warp Mode" function is available, which is very useful to speed up the execution time of our listings.

We would like to thank all the passionate readers of RMW that sent us the results of their works. In this issue we welcome other classic algorithms provided by **Armando Pavese** for C64 and Vic-20 (obviously "doped" with Simons' Basic and SuperExpander VIC-1211M expansions) and a particular solution for Commodore Plus/4 and 128 to draw the Ulam Spiral, based on the arrangement of prime numbers, that has been proposed by **Giuseppe Mignogna**.

It is worth spending a few words about this particular algorithm and the way it was implemented by Giuseppe. The Ulam spiral is a diagram consisting of the sequence of positive integers arranged in a precise order, that is in a spiral. In this particular scheme, the position of the prime numbers reveals a very interesting graphical pattern that is still being studied today. The idea came from the Polish mathematician Stanislaw Ulam, who scribbled the integer numbers in a spiral (see fig. A) and realized that the prime numbers tended to be arranged along some diagonals of the diagram. The fact that the prime numbers are arranged along the diagonals is not in itself a particular fact because obviously all prime numbers except 2 are odd numbers. What was of interest to Ulam and to a long series of mathematicians who over time approached the problem is that prime numbers tend to arrange themselves along certain diagonals and not others. Some rigorous studies have established that the concentration of prime numbers is actually greater on certain diagonals. From a visual and aesthetic point of view, which is what interests us most in this column, the algorithm that generates the spiral of Ulam produces a nice graphic effect if you color the prime numbers and leave "turned off" the non-primes. You can admire the effect in the Plus/4 and 128 versions, and the BASIC listing published is executable on both 8-bit Commodore machines without changes. For the sake of completeness and to honor his work, we say at once that Giuseppe has sent us the listing that you see in these pages, but also a version that overcomes, in the manner of practical programmers of the past, the problem of the slowness of 8-bit computers in computing the N useful primes. The listing presented here actually calculates the prime numbers needed for coloring, while the other version includes a preloaded list of prime numbers up to 32159 using DATA lines. Obviously the second listing, which we do not publish because of its relative length, draws the spiral and completes it in the high resolution screen of the Plus/4 and C128 in a much shorter time. The program is available to readers on request.

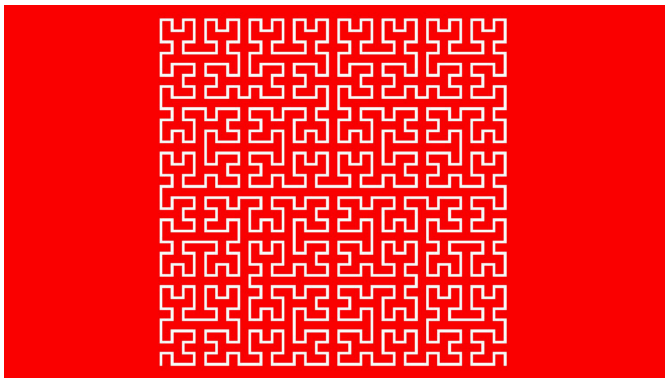
And now, buck up and let's get our hands on our keyboards! Here's another selection of fractals and geometric figures for your and our beloved 8-bits!



12. DRAGON 32 - Lorenz attractor (XY plane)

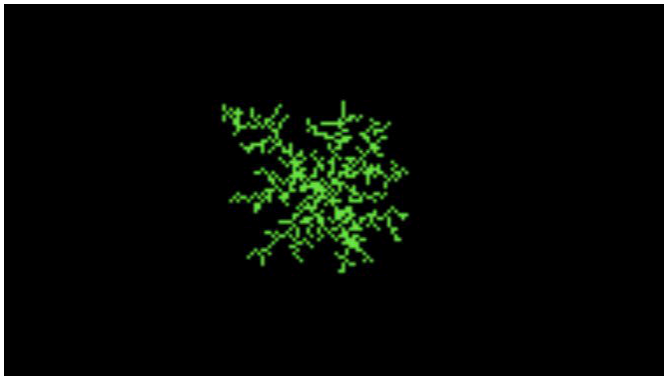
```
10 PMODE 4,1:SCREEN 1,1
20 COLOR 0,1:PCLS
30 R=10:S=28:B=8/3:D=.002
40 X=.1:Y=0:Z=0
50 FOR I=1 TO 16000
60 PSET(127+5.7*X,95-3*Y)
70 XX=X+D*R*(Y-X)
80 YY=Y+D*(X*(S-Z)-Y)
```

```
90 Z=Z+D*(X*Y-B*Z)
100 X=XX:Y=YY
110 NEXT
120 GO TO 120
```



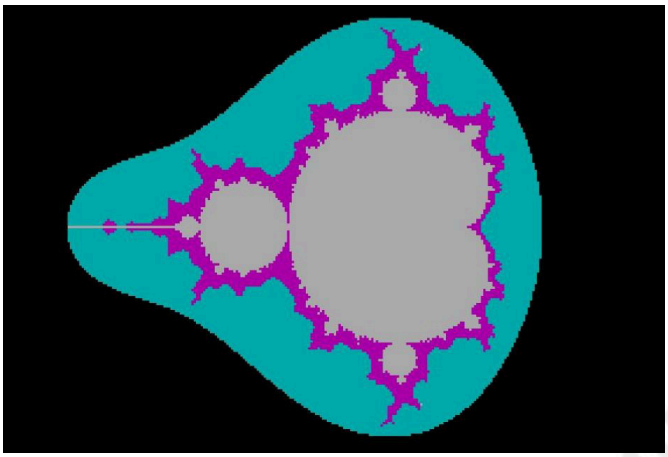
13. ORIC-1 - Hilbert curve

```
10 HIRES:PAPER 1
20 DIM S(50)
30 CURSET 57,165,1
40 S(0)=5:S(1)=1:SP=2:H=4:V=0
50 GOSUB 70
60 END
70 SP=SP-2:N=S(SP)-1:A=S(SP+1)
80 IF N=-1 THEN RETURN
90 S(SP)=A:S(SP+1)=N:S(SP+2)=-A
100 S(SP+3)=-A:S(SP+4)=N:S(SP+5)=A
110 S(SP+6)=N:S(SP+7)=A
120 S(SP+8)=-A:S(SP+9)=N:S(SP+10)=-A
130 S(SP+11)=A:SP=SP+12
140 GOSUB 170:GOSUB 70:DRAW H,V,1
145 GOSUB 170:GOSUB 70:DRAW H,V,1
150 GOSUB 70:GOSUB 170:DRAW H,V,1
155 GOSUB 70:GOSUB 170
160 RETURN
170 SP=SP-1:A=S(SP):T=-A*H:H=A*V:V=T
180 RETURN
```



14. ZX Spectrum Basic - Brownian tree

```
10 INK 4: PAPER 0: BORDER 0: BRIGHT 1: CLS
20 RANDOMIZE
30 PLOT 127,87
40 FOR i=8 TO 32 STEP 4
50 FOR j=1 TO 4*i
60 LET x=127+INT (2*i*RND) -i
70 LET y=87+INT (2*i*RND) -i
80 IF POINT (x,y) THEN GO TO 60
90 PLOT x,y
100 LET n=0
110 FOR a=x-1 TO x+1
120 FOR b=y-1 TO y+1
130 LET n=n+POINT (a,b)
140 NEXT b
150 NEXT a
160 IF n>1 THEN GO TO 250
170 PLOT INVERSE 1;x,y
180 LET x=x+INT (3*RND)-1
190 LET y=y+INT (3*RND)-1
200 IF x>=127+i THEN LET x=127-i+1: GO TO 220
210 IF x<=127-i THEN LET x=127+i-1
220 IF y>=87+i THEN LET y=87-i+1: GO TO 90
230 IF y<=87-1 THEN LET y=87+i-1
240 GO TO 90
250 NEXT j
260 NEXT i
```



15. IBM PC 8088 BASIC, 1981 - Mandelbrot set

(The listing can also be compacted into a single line!)

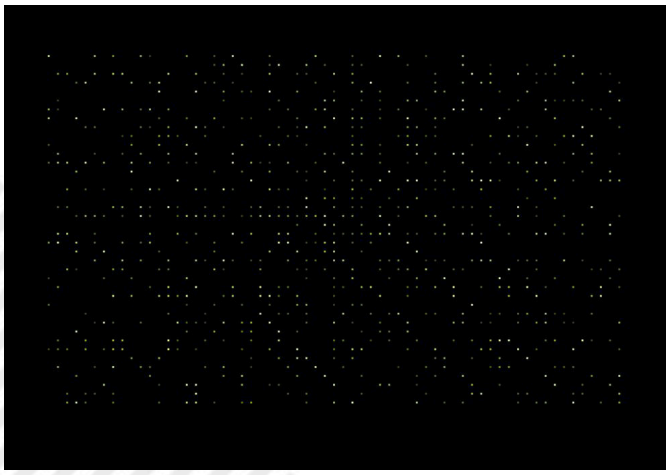
```
10 key off:screen 1
20 for i=-160 to 160
30 for j=-96 to 96
40 x=i/76:y=j/76
50 a=0:b=0:k=0
60 while a*a+b*b<4 and k<200
70 c=a*a-b*b+x:b=2*a*b+y:a=c:k=k+1
80 wend
90 pset(185+i,100+j),0-(k=200)-(k>=10)-(k>=4)
100 next
110 next
```

```
120 while -1:wend
```

16. VIC-20 + SUPER EXPANDER JP VIC-1211M (6502, 1981) - Barnsley fern

(arranged by Armando Pavese)

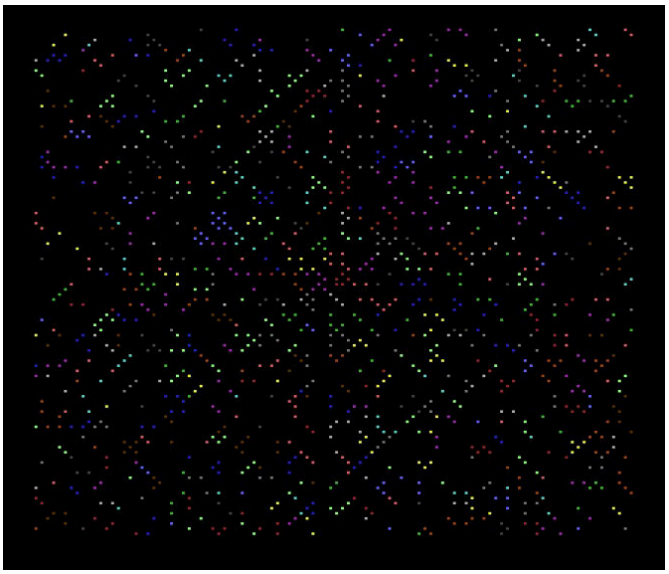
```
10 hires0:setc6,1,7
20 x=0:y=0
30 fori=-3500to3500
40 r=rnd(1)
50 ifr<.01thenx=0:y=.16*y:goto90
60 ifr<.86thent=.85*x+.04*y:y=-.04*x+.
85*y+1.6:x=t:goto90
70 ifr<.93thent=.2*x-.26*y:y=.23*x+.
22*y+1.6:x=t:goto90
80 t=-.15*x+.28*y:y=.26*x+.24*y+.44:x=t
90 plot(25*y)/1.45,90-25*x
100 nexti
110 wait198,1
120 text:end
```



17. C16-Plus/4 (first image) & C128 (second image) - Ulam spiral

(by Giuseppe Mignogna - basic version)

```
10 color0,1:color4,1:graphic1,1
20 x=160:y=100:c=0:d=1:p=5:np=1
30 fori=1to2
40 forj=0toc
50 gosub200
55 color1,8,l:l=l+1:ifl=8thenl=0
60 ifr=1thendraw1,x,y
70 ifd=1thenx=x+p
80 ifd=2theny=y-p
90 ifd=3thenx=x-p
100 ifd=4theny=y+p
120 np=np+1
130 next
140 d=d+1:ifd=5thend=1
150 next
160 c=c+1
170 goto30
180 rem sub verifica se np = nro primo
190 rem risultato in r (0=false,1=true)
200 ifnp=1thenr=1:goto300
210 r=0:dv=3
220 ifnp/dv=int(np/dv)thenr=0:goto300
230 dv=dv+2
```



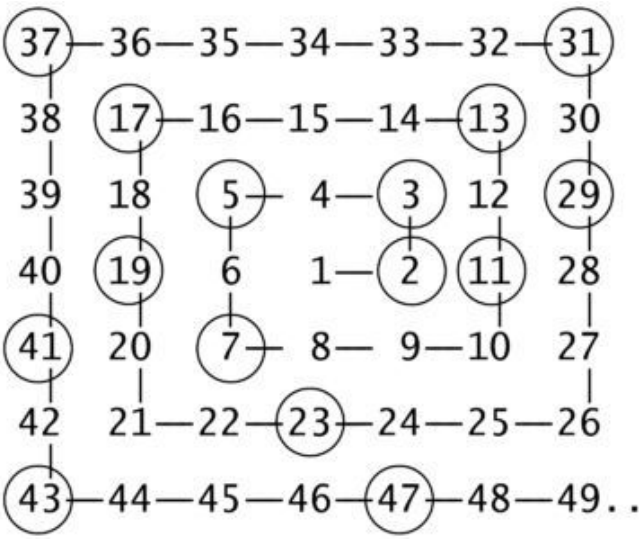
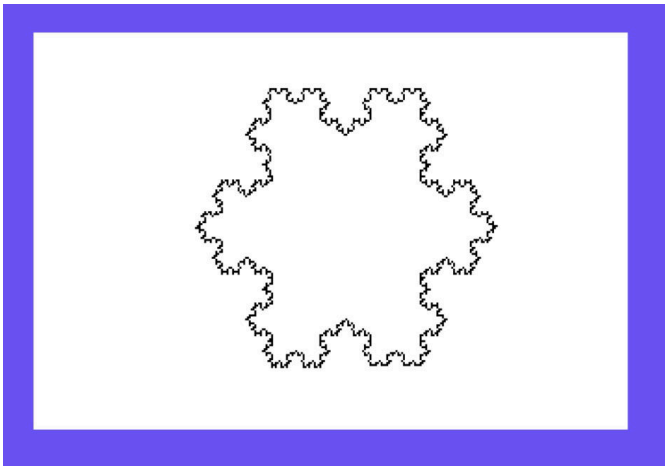


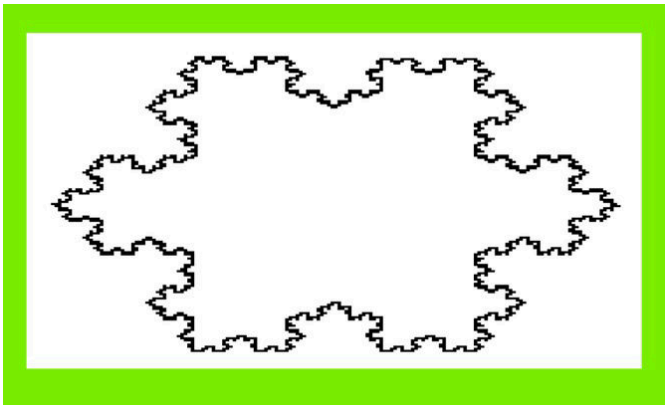
Fig. 1 - The Ulam spiral, was discovered by Polish mathematician Stanislaw Ulam in 1963, who noticed that prime numbers tended to align along the diagonals of the spiral.

```
240 ifdv<=sqr(np)then220
250 r=1
300 return
```



18. C64 + Simons' BASIC - Il fiocco di neve di Koch
(con il contributo di Armando Pavese)

```
10 hires0,1
20 dims(20)
30 a=0:d=π/3:x=128:y=168
40 fori=1to3
50 a=a+2*d
60 s(0)=140:sp=1
70 gosub100
80 next
85 wait198,1:nrm
90 end
100 sp=sp-1:l=s(sp)
110 plotx,y,1
120 ifl<=5thenh=l*sin(a):v=l*cos(a):
linex,y,x+h,y+v,1:x=x+h:y=y+v:return
130 l=l/3
140 forp=sptosp+3:s(p)=l:next
150 sp=sp+4:gosub100:a=a-d:gosub100
160 a=a+2*d:gosub100:a=a-d:gosub100
170 return
```



19. VIC-20 + SUPER EXPANDER JP VIC-1211M (6502, 1981) - The Koch snowflake
(with the contribution of Armando Pavese)

```
10 hires1:setc1,2,6
20 dims(20)
30 a=0:d=./3:x=48:y=148
40 fori=1to3
50 a=a+2*d
60 s(0)=140:sp=1
70 gosub100
80 next
85 wait198,1:text
90 end
100 sp=sp-1:l=s(sp)
110 plotx,y
120 ifl<=5thenh=l*sin(a):v=l*cos(a):
x=x+h:y=y+v:plottox,y:return
130 l=l/3
140 forp=sptosp+3:s(p)=l:next
150 sp=sp+4:gosub100
155 a=a-d:gosub100
160 a=a+2*d:gosub100
165 a=a-d:gosub100
170 return
```

The third part of this column stops here, but the success that we have found so far lets us hope that we will have other interesting listings to offer you in the future, in order to bring you back to BASIC programming and to your beloved 8-bit microcomputers.

So, if you have any interesting ideas or hints for the column, please send them along with a few lines of information and/or instructions to our official email box: RetroMagazine.Redazione@gmail.com.

All submissions will be viewed and given due consideration and, as always, the best and most intriguing will get the honour of publication in future issues of the magazine.

C128 in 80 column mode - part 2

by Gianluca Girelli

In the previous article, published on RMW #34-IT (RMW#12-EN), we introduced the 8563 chip (also called VDC) that is that part of the system dedicated to the management of the screen in 80-column mode.

We learned that, unlike the VIC (chip 8564) used for the standard 40-column mode, it is not possible to access the VDC directly with the PEEK and POKE commands of BASIC but it is necessary to use machine language and refer to two specific memory locations: 54874 (address register, $D600) and 54875 (data register, $D601).

Through them, in fact, we will have access to the 37 registers of the chip and their manipulation will allow us to act on its 16KB RAM, independent of the system RAM. We have also seen how to write data to memory through these registers: you have to enter in 54874 the number of the register you want to interact with, and when bits 6 and 7 will take the value "1" (managed by the system), you can store a value in 54875. This will force that value to be passed into the register pointed to by 54874. This mechanism was useful for reprogramming the basic C128 character font and obtaining a "Japanese-like" one. The associated code has been commented on in the above mentioned issue 34, to which we refer for further information.

Functioning of the I/O locations is the same also in the case of the 8563 register read operation: simply, when bits 6 and 7 go to "1", just read the contents of 54875 which will contain the desired register value and "pointed" from location 54874.
Let's see how.

The following code, taken from the book "Commodore 128 Oltre il Manuale", makes use of the BASIC "USR" function.

This command, used to implement a user-defined function by expanding the set of routines of the C128, when called with a syntax such as "W=USR(K)" behaves as follows:
- places the value of "K" in floating point accumulator number 1 (named "FAC1");
- transfers the control to a machine language routine written by the user and stored from the address defined by the (decimal) locations 4633 (low byte) and 4634 (high byte) [Note: on C64 the locations have decimal address 785 and 786;]
- once the LM routine has been executed, it places the result in FAC1 before returning to the calling program;
- the calling program places the result in the variable "W" (in the case of the example) or prints it directly on the screen if an instruction of the type "PRINT USR(K)" is used.

Let's start by reporting the complete code of the machine language routine, the first part of which (about writing registers) was commented on issue 34:

```
F0B00 LDY   #$00
F0B02 STY   $FF00
F0B05 STA   $D600
F0B08 BIT   $D600
F0B0B BPL   $0B08
F0B0D STX   $D601
F0B10 RTS
F0B11 LDA   #$00
F0B13 STA   $FF00
F0B16 JSR   $AF0C
F0B19 LDA   $16
F0B1B STA   $D600
F0B1E BIT   $D600
F0B21 BPL   $0B1E
F0B23 LDY   $D601
F0B26 LDA   #$00
F0B28 JMP   $AF03
```

As reported in the previous paper, we chose to store the routine in the cassette buffer, which starts at location 2816, (hexadecimal 0B00). For those unfamiliar with C128, we report the procedure for inserting programs into memory using the machine language "monitor":
- type the BASIC command "MONITOR";
- insert the first line by typing "A F0B00LDY #$00", followed by "RETURN". The editor will translate the code directly into machine language by calculating the memory occupancy in bytes and entering the correct next line number.

Note that the address "F0B00" indicates that we are writing to memory in "bank 15", "F" in hexadecimal (see the bibliography for more information);

- Once all the code has been entered, press "RETURN" to exit the editing mode, save the program with the command S "RR8563" 8 F0B00 F0B2B then exit the monitor by pressing "X" (RETURN). Note that the last byte of the code is inserted in the location F0B2A, but to store it in a binary file you must increase this address by 1 byte.

Omitting what happens in the program between locations F0B00 and F0B10, already examined above, let's try to understand how the reading process works assuming we want to read the contents of register number 3 of 8563:
- we take the start address of the read subroutine (0B11, the bank number is not important in this case) and separate it into its components "0B" (decimal 11) and "11" (decimal 17). These values, in the form low byte and high byte, should be set with POKE 4633.17 and POKE 4634.11 to tell the "USR" routine where to jump;

- we call the routine with the command "USR(3)". The value "3" (this is the register we want to read from) will be inserted into the FAC1 accumulator;

- at this point the LM program invokes the system routine "GETADR" (JSR $AF0C), which converts the value of FAC1 into a two-byte integer. The low byte is stored in decimal location 22 ($16), the high byte in the next decimal location. We are only interested in the first one because it represents the register number with which we are interacting, so the instruction "LDA $16" takes the register number (3) and puts it in the I/O location $D600;

- once bits 6 and 7 of $D600 go to 1, we can automatically read the contents of register "3" by taking it from the next location. This value will be loaded into the register of chip 6502 named "Y";

- after loading the accumulator with the value "0" (LDA #$00) to reconstruct a two-byte address, we invoke the system routine "GIVAYF" (JMP $AF03). The GIVAYF routine transforms the number of two bytes found in "A" and "Y" ("A" contains the high byte and "Y" the low byte) into a value to be re-entered in FAC1, from which it is extracted by assigning it to a variable (for example with the pre-mentioned syntax such as "W=USR(3)") or by printing it on the screen, as we can see in the following BASIC code.

```
1 rem reg8563
3 rem writing into the chip registers 80 columns
7 bload"rr8563",d0,u8,b15,p2816
13 poke 4633.17:poke 4634.11
15 print "registers content 8563":print
17 for k=0 to 36
19 : print "register",k,usr(k)
21 next k:print
25 print "you want to write in the registers"
27 input "of chip 8563? (s/n)";r$
29 if r$="n" then stop
31 input "write in register (0/36) ";n
33 input "the data: ";d
37 sys 2816,n,d
39 goto 25
```

The program takes care of loading into memory the previous LM routine typed and saved through the machine language "MONITOR" (line 7). Then, after setting the pointers to the routine (line 13), the contents of all registers of the 80-column chip are printed to the screen. Finally, the code allows us, if we want, to write some values in a register of our choice and check what happens.

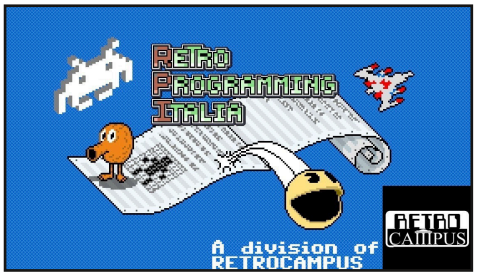
Now that we have learned how to read and write to the chip's memory, we have a whole new set of possibilities in front of us, including fast memory block transfers or doing (monochrome) 80-column graphics. As you probably already know, the BASIC graphics primitives of the C128 only work in 40-column mode, but now that we've figured out how to access the VDC screen we can write our own routines to have fun with.

So see you in the next issue of RMW to explore new opportunities!

Bibliography:

"Commodore 128 Oltre il Manuale", Rita Bonelli, Gruppo Editoriale Jackson Giugno 1986
"C128 Programmer's reference guide", Commodore Business Machines, Inc., Bantam Book February 1986
"Mapping the Commodore 128", Ottis R. Cowper, COMPUTE! Publications,Inc 1986
"Commodore 128 - Manuale di Sistema", Commodore Italiana S.p.A., 1985

How to create a BASIC game for Commodore 64 - part 1



by Felice Nardella (aka Kimono)

This tutorial is intended for those who, like me, since I was a child, has always dreamed of creating a videogame for the legendary Commodore 64, but has never been able to realize it for lack of sufficient knowledge of BASIC and especially notions, at least basic, on the programming of graphics and sounds of the C64 itself, which I will deal more in this Tutorial.

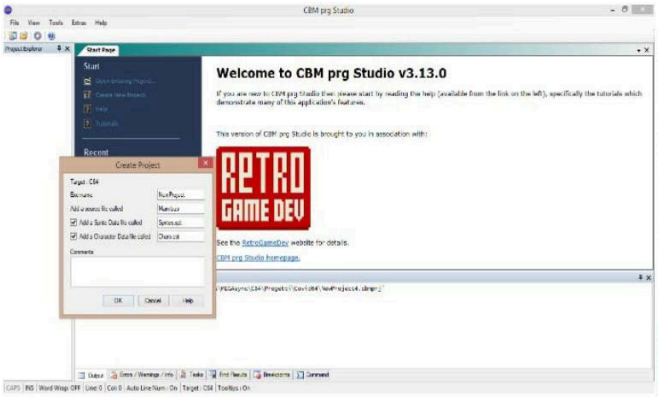
The BASIC v2 language, notoriously integrated into the machine, is in fact quite difficult, especially when you want to program with sprites and/or sound effects, given the total lack of specific commands, compared to other more advanced versions of Basic.

The total lack of these commands makes it necessary to use the memory locations directly by means of the POKE instruction, entering the appropriate value in the various registers, which must be known in a more or less thorough way, to be able to program a game that is at least decent. Another big handicap of Basic is its known slowness in comparison with the Assembly language (ASM); slowness mainly due to the fact that the various Basic instructions and commands must be first translated by the so-called Basic Interpreter, in machine language, language that is, in fact, the only one directly understandable by the machine.

Nevertheless, the slowness of Basic, in cases where it is needed, can be remedied in two ways:
1) Try to optimize the Basic code as much as possible, following some rules of thumb that I will describe in the course of this Tutorial, in order to make its execution faster.
2) To compile the Basic code by means of compilers, that are programs that try to translate all the Basic code, directly in machine code.

In this Tutorial we will deal with all the main arguments to be able to create a game in Basic v2 for C64 and finally we will create one, ex novo, as an example.

What you need to start In the 80's we used to use the C64 directly to write programs and games, nowadays instead, with the advancing of years and technology, when we want to write programs of a certain length, the choice is almost forced to use modern IDEs, such as the famous CBM prg Studio, that you can download from this link: https://www.ajordison.co.uk/download.html and that will facilitate our work in game programming.

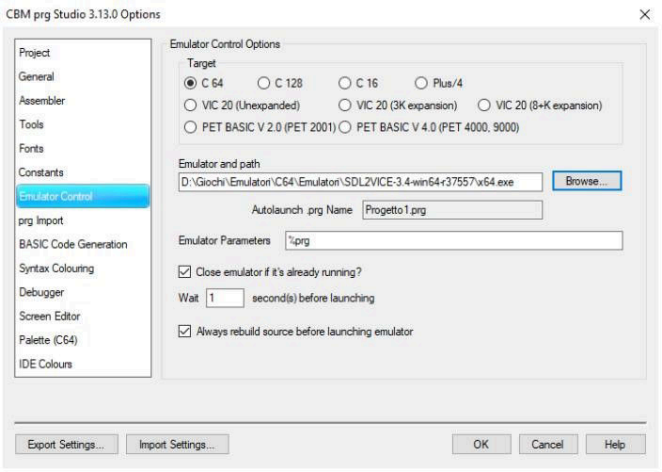


Alternatively, you can use others, such as C64 Studio: https://www.georgrottensteiner.de/en/index.html although in this Tutorial I will only deal with the first one (CBM prg Studio).

CREATE A NEW BASIC PROJECT ON CBM PRG STUDIO

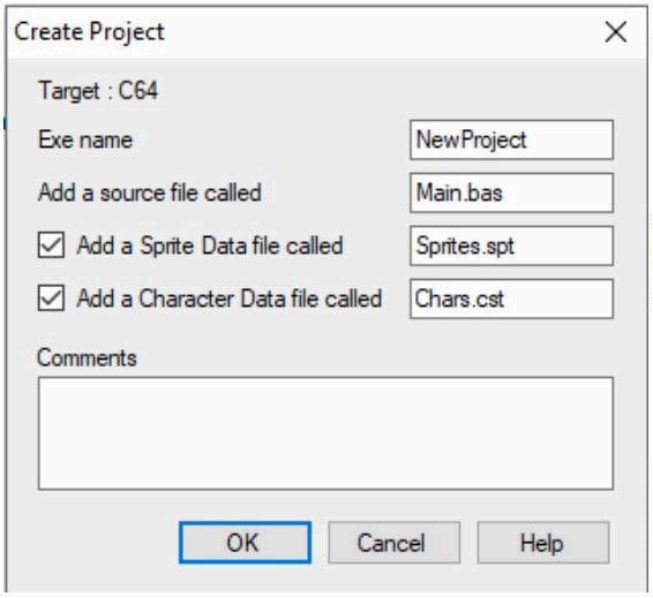
After downloading the IDE "CBM prg Studio" and installing and starting it for the first time, we need to set some important options, including the default directory where we save our new projects.

Also it is possible to connect our C64 emulator to the IDE, by going to TOOLS -> Options... -> Emulator Control



and entering the full path, where the executable (exe) file of the emulator is located, in the "Emulator and path" field.

The emulator I normally use is VICE; if you don't have it, you can download it from here:
https://viceemu.sourceforge.io/index.html#download



Then go to FILE -> New BASIC Project and a window like this will open:

In the field "Exe Name" we will have to insert the name of our project for ex. "Project1"; in the field below we will have to insert the name of the file in Basic that then, once finished, will be compiled, creating in this way the .prg file that will be possible to execute directly on the emulator. The two check-boxes below we can leave: the first one is for adding the possibility to insert one or more Sprites, the other one is for modifying the C64 characters.
Then we press OK: in this way we have created a new Basic Project and we have personalized it with a new name of the project and of the file that will contain our program in Basic.

WHAT IS AND WHAT ISN'T THERE

In this Tutorial, as it is specified in the title, I will deal essentially with the programming of graphics and sounds with Basic v2, in order to write a complete game.

I won't deal with the basics of the programming in Basic v2, for which you refer to the various manuals in circulation on the Web.

CHAPTER 1:

SCREEN MEMORY

The C64 screen is essentially a rectangle consisting of 25 rows (0 to 24) and 40 columns (0 to 39). It is possible

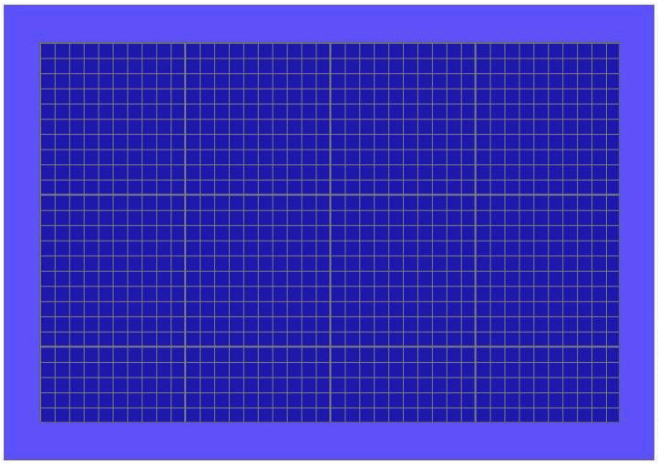


Fig. 1

to divide this rectangle into 1000 (25 x 40) boxes (see Fig. 1) in each of which you can place any Petscii character, be it a letter, a number, or a graphic symbol. Each box corresponds to a memory address according to the scheme shown in Fig. 2 (next page).

The numbering begins with the square in the upper left corner (1024) and ends with the square in the lower right corner (2023). Note that the boxes belonging to the same column differ from each other by 40 units. All the characters on the keyboard correspond to a number between 0 and 127. If you add 128 to the above numbers, you have the same characters, but in reverse.
To be able to see all the characters and codes (Screen Code) corresponding to them, you can use CBM prg Studio, by clicking on the TOOLS -> SCREEN EDITOR menu, a screen similar to the one in figure 3 will open.

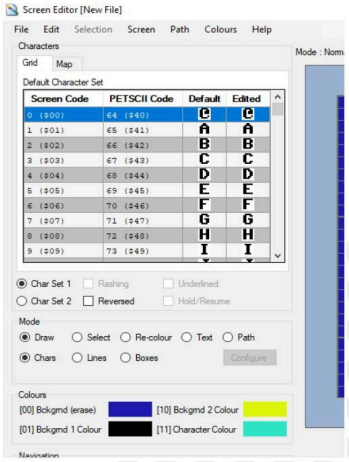


Fig. 3

If now, for example, we want to make any of the Petscii characters appear on the screen, we can use the instruction:

```
POKE A, B
```

Where A indicates the address corresponding to the box in the rectangle and B represents the code of the

RIGHE	COL. 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	
0	1024																																								1063
1	1064																																								1103
2	1104																																								1143
3	1144																																								1183
4	1184																																								1223
5	1224																																								1263
6	1264																																								1303
7	1304																																								1343
8	1344																																								1383
9	1384																																								1423
10	1424																																								1463
11	1464																																								1503
12	1504																																								1543
13	1544																																								1583
14	1584																																								1623
15	1624																																								1663
16	1664																																								1703
17	1704																																								1743
18	1744																																								1783
19	1784																																								1823
20	1824																																								1863
21	1864																																								1903
22	1904																																								1943
23	1944																																								1983
24	1984																																								2023

Fig. 2

corresponding character.

If for example we write: `POKE 2023,0`

a @ (character corresponding to 0) would appear in the lower right corner (Fig. 4); while if we write: POKE 1024,128 a reverse @ (0+128) would appear in the upper left corner (Fig. 5).

Fig. 4

Fig. 5

In another region of RAM, there are other addresses that store the color with which the character should be displayed in each box on the screen.

These addresses go from 55296 to 56295; so if for example we wanted to color the snail in the lower right corner in black we would have to write: POKE 56295,0

Where the 0 represents the black color code.

All the color codes can be obtained from the various manuals, or even directly from CBM prg Studio by clicking on the TOOLS -> SCREEN EDITOR menu and then clicking 2 times on one of the 4 colors in Fig. 3; a small window like the one shown in Fig. 6 will appear.

Fig. 6

Through a simple formula, we will be able to place any character anywhere on the screen, knowing only the row

and column.
The instruction with the general formula is as follows:

```
POKE 1024+ROW*40+COLUMN,CHARACTER
```

While to color it we should use this other formula:

```
POKE 55296+RIGA*40+COLUMN,COLOUR
```

For example, if we wanted a little red heart to appear on line 13 and column 0 (remember that column 0 is the first column on the left) we would have to write a little program of only 2 lines and give RUN:

```
10 POKE 1024+13*40+0.83
20 POKE 55296+13*40+0.2
```

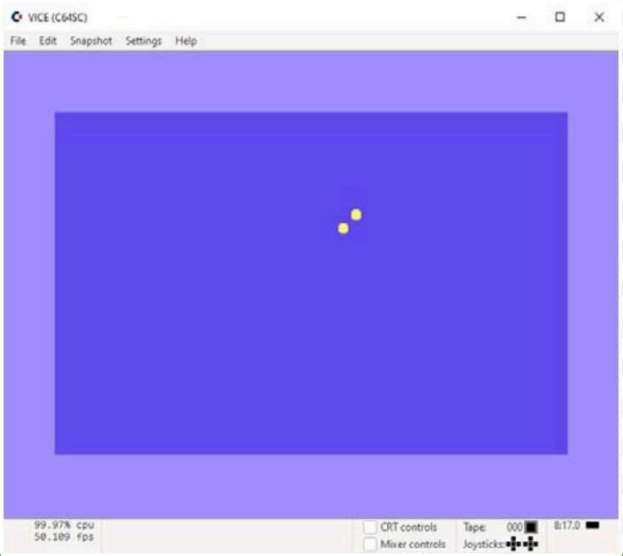
Trick: *to write "on the fly" a small program like the one just seen, but also larger, without going through the IDE, it is usually not convenient to write it directly on the emulator. It is better, instead, to write it (all in lowercase) in a text file (txt) and then do the copy/paste on the emulator.*

A FIRST EXAMPLE: A BOUNCING BALL

Now that we have figured out how to display the characters on the screen, let's try to beat on CBM prg Studio a little program that displays a bouncing ball on the screen. Let's create on CBM prg Studio a new Basic Project and let's call it "Example1". Then rename the file "Main.bas" to "Ball.bas". In this case we can remove the two check marks below, because in this first example we will have neither the need to use sprites, nor the need to change the standard C64 characters. Now we type the following program:

```
10 print"{clear}": rem clears the screen
20 sm=1024:cm=55296: rem initialize variables screen memory and color memory
30 p=81: rem stores the code of the ball character
40 ko=7: rem memorizes the color code of the ball (yellow)
50 ca=32: rem stores the code of the car. space used to delete the ball
60 co=1:ri=1: rem initializes the initial position of the ball
70 dx=1:dy=1: rem initializes the horizontal and vertical increment
80 pp=ri*40+co: rem calculates the initial position of the ball
90 co=co+dx:ri=ri+dy:np=ri*40+co: rem updates the position of the ball
100 if ri>23 then dy=-1: rem check that the ball does not go beyond the line 24
110 if ri<1 then dy=1: rem check that the ball does not go beyond line 0
120 if co>38 then dx=-1: rem check that the ball does not go beyond column 40
130 if co<1 then dx=1: rem checks that the ball does not go beyond column 0
140 poke sm+np,p: poke cm+np,ko: rem prints the ball and colors it on the screen
150 poke sm+pp,ca: pp=np: rem delete ball and update initial position
160 goto 90: rem restarts the cycle from line 90
```

As you can see, each line has been duly commented (comments can also be omitted when copying the code, just read them and understand what the program does, line by line). Then press the PRG button at the top. If no mistakes were made, you should find your PRG file ready to run inside the folder you have chosen as default for projects. If you have already connected VICE to the IDE via the settings, just press the button immediately to the right (->) to run it directly on VICE.



Musica Maestro SID!

by Felice Nardella (aka Kimono)

We all know that a SID file cannot be played on a real C64 or an emulator... But if you want to listen to it, how would you do it?
Well, there are several ways to do this: a rather simple one is to directly transform the SID into an executable PRG file, via the PSID64 tool: http://psid64.sourceforge.net/.

But we're used to getting our hands dirty with code, so I'll present here a short program, which will serve our purpose.

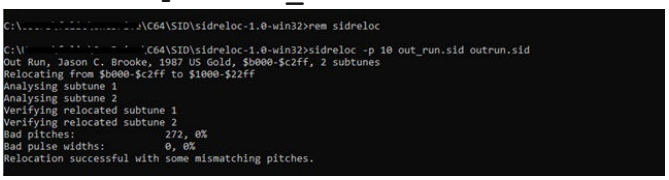
It's good to specify that it's not possible to do this from Basic, but you must use Assembly; in this case I used Basic only to load the ASM code (and the SID) in memory and execute it. This method will work with almost all SID files (they can be downloaded, for example, from https://hvsc.c64.org/), but not with all of them.

In particular, SIDs that have an INIT ADDRESS at location $1000 (4096 dec) will work. If the SID has a different address, this method will not work, but all is not lost! There is a tool called SIDRELOC, downloadable from a very good site, where you can get various useful information about SIDs (http://www.linusakesson.net/software/sidreloc/index.php), which is able to relocate the SID in the desired memory area (a version of SIDRELOC compiled for Windows can be downloaded from CSDB: https://csdb.dk/release/?id=109000).
Specifically, here I tried to run the SID file of the famous game OUTRUN, which you can find here: https://csdb.dk/sid/?id=4154.

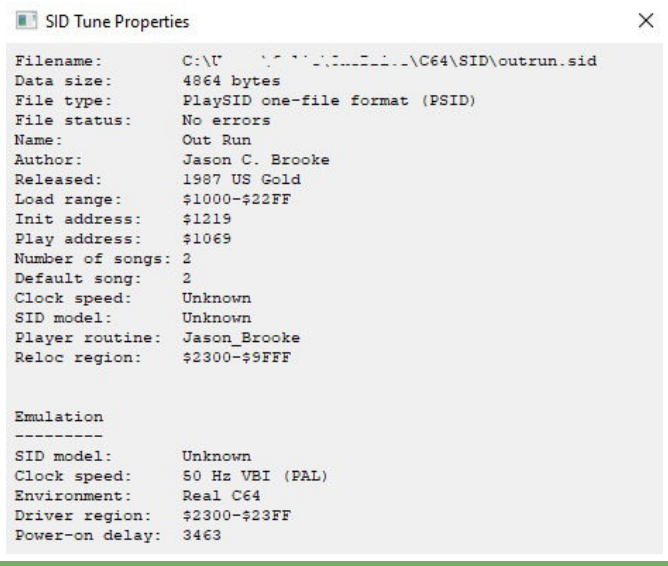
Executing such a file, through the Sidplayfp tool (https://csdb.dk/release/?id=210057), it is necessary to look among the properties of the file (through the menu FILE -> PROPERTIES) and especially focusing on the following values:
- Init address: $B219
- Play address: $B069
- Number of songs: 2

You can see that this SID has an Init address different from $1000 and that it has 2 tracks.
Then we must try to relocate it as close as possible to the address $1000, using the aforementioned SIDRELOC, by typing from the command line:

```
sidreloc -p 10 out_run.sid
```


\C64\SID\sidreloc-1.0-win32>rem sidreloc
\C64\SID\sidreloc-1.0-win32>sidreloc -p 10 out_run.sid outrun.sid
Out Run, Jason C. Brooke, 1987 US Gold, $b000-$c2ff, 2 subtunes
Relocating from $b000-$c2ff to $1000-$22ff
Analysing subtune 1
Analysing subtune 2
Verifying relocated subtune 1
Verifying relocated subtune 2
Bad pitches: 272, 0%
Bad pulse widths: 0, 0%
Relocation successful with some mismatching pitches.

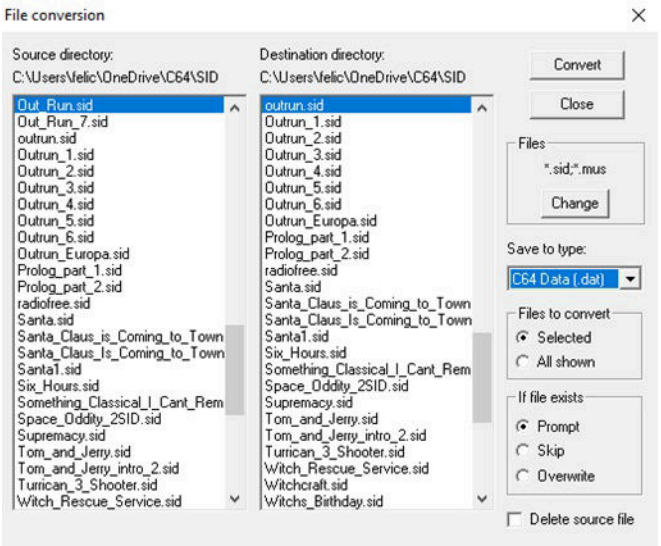
In this case, "Bad pitches" errors can be safely ignored. Looking now among the properties of the new obtained file (outrun.sid).
We will have the following parameters:



- Init address: $1219
- Play address: $1069

These two values will need to be included in our code.

Before executing the Sid, however, it is necessary to transform it in a .DAT file, always through the tool Sidplayfp, going on FILE -> CONVERSION, selecting on "Save Type" the option "C64 Data (.dat)" and then clicking on CONVERT. We will get the file "outrun.dat" that we will use later. This transformation is necessary because SID files have an initial header of $7c (124 dec) bytes that, in this case, must be removed before they can be placed in the C64 memory. You can also manually delete this part using a good hex editor, such as WINHEX (http://www.winhex.com/winhex/), see fig. 1.



This the code I used:

```
10 poke53280,0:poke53281,0:me=49152
20 readop:ifop<0then90
30 pokeme,op:me=me+1
40 goto20
90 ifa=0thena=1:load "outrun.dat",8,1
95 poke53280,15:sys49152
100 date 120: rem six
110 data 169.0: rem lda #0
120 data 32,25,18: rem jsr $1219 - init addr
130 data 32,105,16: rem jsr $1069 - play addr
140 data 173,18,208: rem lda $d012
150 data 201,250: rem cmp #250
160 data 208,249: rem bne l140
170 data 173,1,220: rem lda $dc01
180 data 201,255: rem cmp #$ff
190 data 240,239: rem beq l130
200 data 169.0: rem lda #0
210 data 141,24,212: rem sta $d418
220 data 88: rem cli
230 data 96: rem rts
240 date -1
```

In the meantime, let's remember the name of the DAT file we have just obtained to be written in line 90 (as indicated in the listing).

In the DATA of the lines 120 and 130 we will have to insert, respectively, the values of Init Address and Play Address. For those who have difficulty in performing this step, I will briefly explain how to proceed:

- the value $1219 (in hexadecimal) is expressed in binary like this:

0001 0010 0001 1001

This is a 16-bit value, which you need to decompose into two 8-bit values.

- The first value to be entered is the low byte (right) which, transformed into a decimal is:

0001 1001 = 25

- The second is given by the high byte (left):

0001 0010 = 18

- These values should be entered in the DATA of line 120, after the value 32 (representing the JSR OpCode in decimal).
- The same operation should be done for the Play Address values.

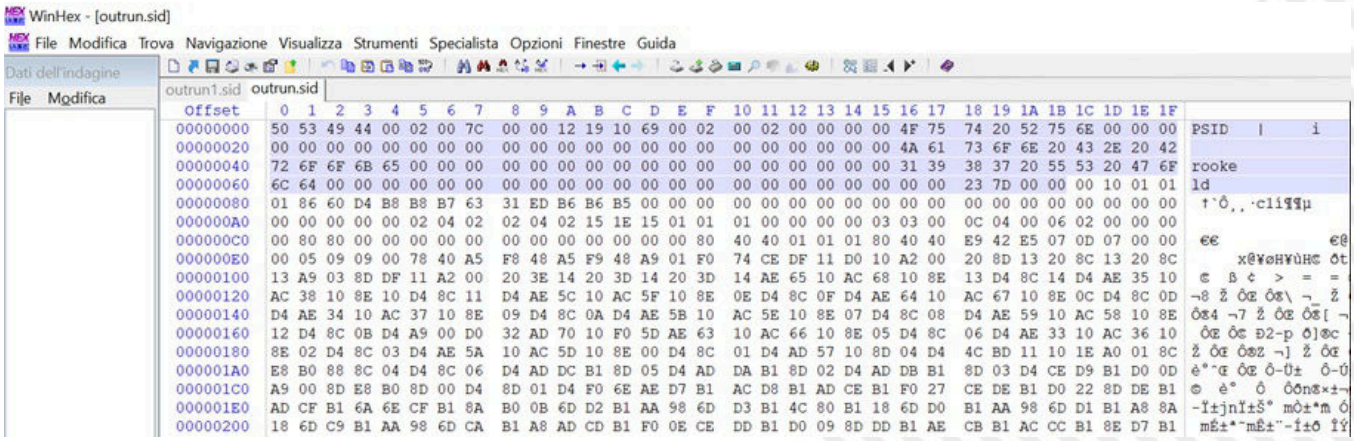


Fig. 1 - WINHEX Hex Editor

To change the track, in case there are more than one (in our case we have 2), just go to change the value in line 110 after the digit 169 (OpCode LDA in decimal), in this case from 0 to 1, or you can also directly change the value of the location 49153 through a POKE:
POKE 49153.1
and re-run the code with RUN.

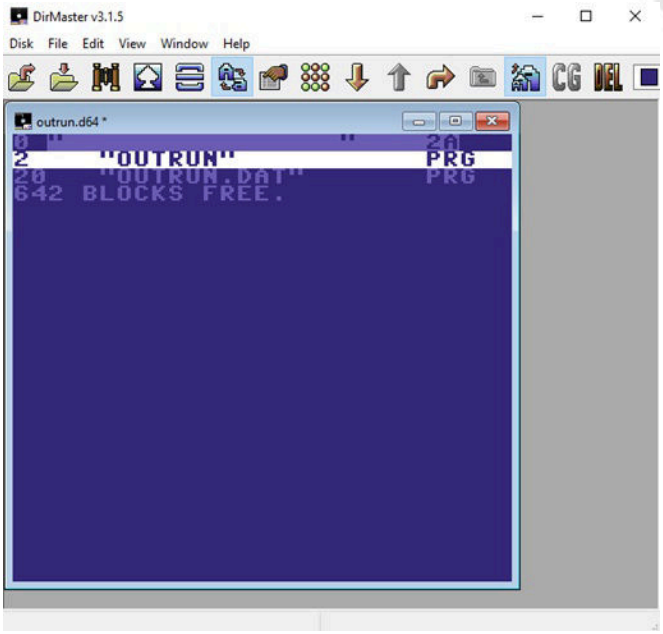
The location $DC01 (56321 dec) indicated in line 170, is the one that contains the state of the joystick in port 1, but also of some keys of the keyboard (in particular SPACE, CTRL, 1, 2, ARROW). If one of these keys is pressed during execution, or if the joystick in port 1 is moved, the program exits. You could have entered another location to control, such as the joystick in port 2 ($DC00); in that case the keys to press to exit the program would be SPACE + F1/F2, SPACE + Z, SPACE + B, SPACE + C, SPACE + M, or the movement of the joystick in port 2. The value in line 180 (control) should have changed from 255 to 127 ($7F).

Lines 200 and 210 before exiting, reset the volume of the SID, acting on Register $D418 of the audio chip, while the last DATA of line 240 (-1) is simply a control byte for exiting the loading of the Assembly routine into memory, starting at location 49152 ($C000).

The resulting PRG file must be enclosed in a D64 file together with the DAT file, using the very useful DIRMASTER tool (https://style64.org/dirmaster). The PRG file, inside the D64, must necessarily precede the DAT file, if you want it to be started automatically by the emulator.

The resulting D64 file can then be easily run on the emulator or copied to floppy (or other media) to be started from the real C64.

Enjoy listening!



Useful Links

We report again the links presented in the article, for your convenience.

PSID64:
- http://psid64.sourceforge.net/

HVSC - High Voltage SID Collection:
- https://hvsc.c64.org/

SIDRELOC:
- http://www.linusakesson.net/software/sidreloc/index.php

SIDRELOC for Windows:
- https://csdb.dk/release/?id=109000

Sidplayfp:
-https://csdb.dk/release/?id=210057

DIRMASTER:
- https://style64.org/dirmaster

SID of the game OUTRUN:
- https://csdb.dk/sid/?id=4154

*The entire editorial staff of RetroMagazine World thanks and appreciates the work of the Facebook group "***RetroProgramming Italia - RP Italia" - A division of "RetroCampus" Associazione Culturale***
and thanks them for their kind contributions.

We believe that, like other similar groups, they are doing a great job to make known the Italian retrocoding scene in the world.

We invite our readers to collaborate with them in case they have material to share with all the fans of retrocomputing and retrocoding.

Interview with Bas Scheijde, the father of Puzzle Bobble on Commodore 64

by Carlo Nithaiah Del Mar Pirazzini

Puzzle Bobble in C64 version is a gem! The review speaks for itself and we want to specify that to make a software of this level you need knowledge, knowledge and ... knowledge.
We had a chat with Bas Scheijde, developer and creative mind behind the game.

Nith: Hi Bas, Thanks for your time. Tell us a little bit about yourself. When did your work as a retro-developer start?

I am a self educated software developer, My coding skills are mainly 6502 assembly, basic, VB.net and some small knowledge of javascript, C# and C++
My first programming started at age 9 with a commodore PET in basic, but soon we had a c64 at home.
At some point I got hooked up into assembly, but besides some trainer poke findings and a handful of demo's that never really got published. Or better said the group I was part of, never published anything significant.
After the c64 came the Amiga, but I never programmed on it, this one was purely for gaming and making music.
But I always kept my c64 in the closet for later. That later came 5 years ago. I dusted off my c64, but found that all disks were gone, c64 gave a black screen and the drive didn't have a power supply anymore.
So I went on 2nd hand market site and bought myself a nice new in the box old breadbin for very cheap, along with an sd2iec device and got myself up and running again.
At this point I was also starting to make some little programs to try and see if touch devices (tablets phones) could be hooked up to the light pen input and maybe setup something like a nice gui.
Since there are already some nice projects working on this, I decided to check what games would still be interesting to port.
This is how I ended up with puzzle bobble. There wasn't a port made yet and from what I could find, like monster buster, it was nice, but no real port of the original.
So, here was my project, creating a puzzle bobble port.

Fig. 1 - Bas in the kitchen after cooking....
...the cake doesn't stand a chance

Another concept that can be made from the current project is the snes 100 level version. The game is almost the same in regards to the game physics, just more levels and some "special" balls that have specific features when hit.
But that's for later...
In the meantime I also wondered if it would be possible to create a real pseudo 3D racer for the c64 with an acceptable framerate, using different techniques.
I am a big fan of lotus esprit turbo challange on the amiga and thought that such engine could be possible for the c64.
So one year of development of puzzle bobble was halted to work the idea into a proof of concept, that result can be found on my youtube channel.
I hope to continue soon on the racer project as it seems to be quite good in regards to decent framerate/full flexible pseudo 3D road building.
But its still a loooong way from a game concept.

N: Now tell us about Puzzle Bobble and how you started its development.

Puzzle bobble was created by taito in the early 90's and is one of the most popular puzzling games made.

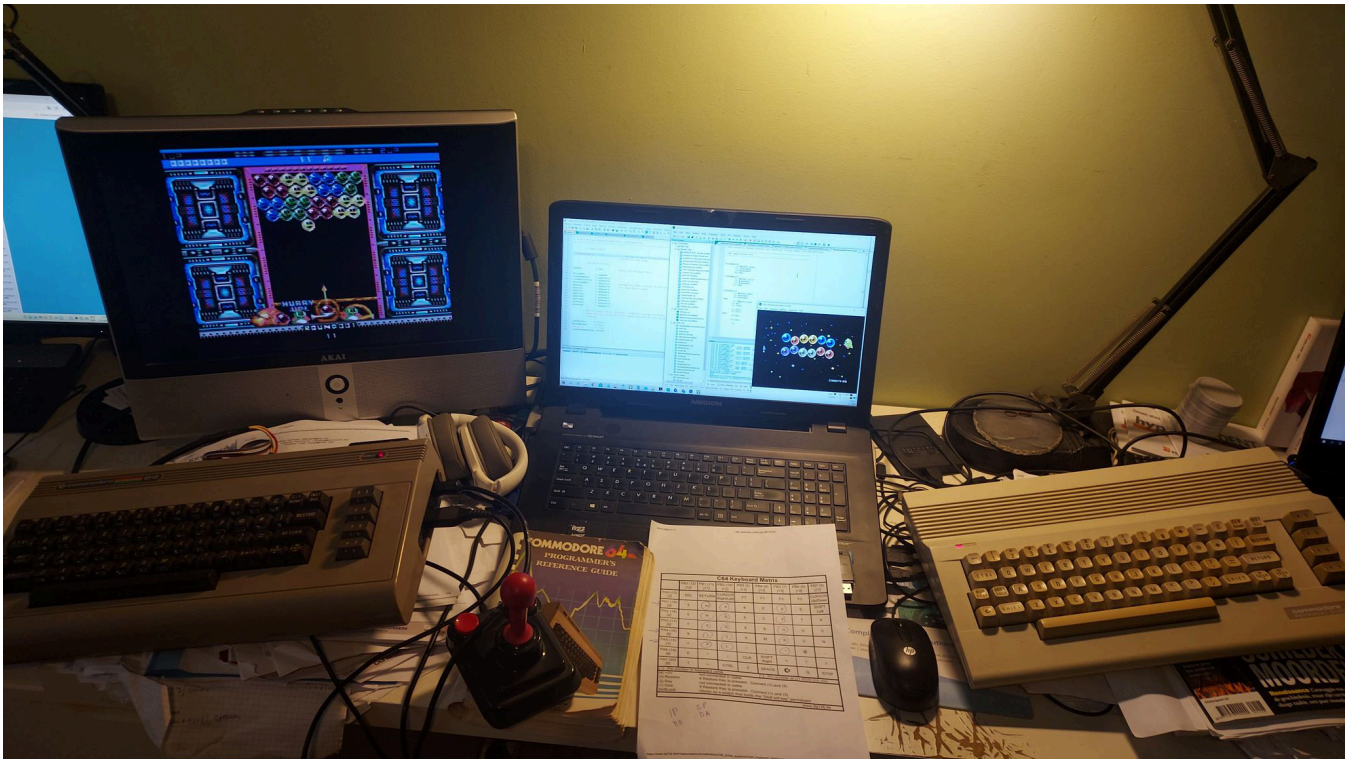


Fig. 2 - The development workstation of Puzzle Bobble by Bas

And when I started the project I also thought it would be fun to create a MAME like port, using the same keys and keep the coin in also in the game.

Obviously this lead to... also porting the intro. Initially a pretty heavy extra task, but also fun to do.

The whole game including the intro is bitmap based, no charsets are used, only bitmap plotting and sprites. To make the animations manageable, I coded an sequencer, which can run through functions in batch order with given parameters.

Initially this was made to setup the level backgrounds, but this was later also very useful for creating the intro animations.

Since there are so many animations happening on screen I also decided to use a sprite overlay on the playfield, in which some animations take place like the falling bastards that pop out of the ballons and the floating points. these animations are also platted into this overlay. In 1 player, 2 overlays on top of each other, in 2 player, each player has its own overlay, giving a slight color clash between points and falling bastards, but I found this acceptable.

In order to manage all animations, I use a 4 page animation array with 255 slots, keeping track and performing all animations outside the IRQ, while the gameplay itself stays inside the IRQ, letting the gameplay being unaffected if a lot of animations happen at the same time.

On the music and sfx, I worked hard together with NM156 to make the most out of it. So I hacked into the goattracker player and directly fire up patterns in a free voice while the music plays on the other 2 voices. In case music is off, the sfx are divided over all 3 voices, for getting even more back from the sfx's. In 2 player, we had to settle with 1 voice for music and each player a dedicated voice. NM156 (frantic freddy 2) did a great job, creating a whole span of patterns for a lot of different balls amount popping or falling as well as recreating all music tunes from scratch. Hend (wild woods) helped me out a great deal with mocking up the graphics, added a lot of great stuff, and kept a few of my original design also in.

Some nice features we added with help of Hend was creating level dedicated press for each 3 levels and fantastic drawing on the end screen pic.

All with all, it was a long time developing, but I am glad it all worked out in the end.

Tips from me for anybody who wants to do similar projects. - make a playable proof of concept, that doesnt need to be good looking, but must work properly for the better part of it. You can always find music or graphics artist that help out, but too many projects end up on the "we will finish this one day" when the point of complexity gets

too big.

- don't hesitate to do big modification ones you notice you took a wrong path in coding, better to change course in time then do it at the end.

- make little side projects to test out stuff and get to know scripts you use from others. In my case I spent a lot of time diving into digi/sample playing, using Lasse's Bitops Fastloader and hacking into the Lasse's goattracker routine to control play speed and play on the fly patterns while the music is playing.

- reiterate your code frequently to optimize in speed or size. Most of the time, the proof of concept is great, but looking back at some code a year later, changing your perspective, might bring you onto ideas to improve old code immensely.

There is another side note I would like to make. I would never have started this project if it wasn't for CBM prg Studio (Arthur Jordison).
Without such an IDE, I probably would have not be able to make this without it.

All tools that were used:
CBM prg studio
Goattraker2
Covert bitops fastloader
Multipaint
Spritepad
Exomizer
Dirmaster

Sites used for info:
Codebase64.org (lots of info and several scripts for direct use)
6502.org (the source for getting details of 6502 assembly)

N: Thank you for all the responses and your valuable time. Before you say goodbye, tell us about your future plans.

Since I only am familiar with the c64/6510 as retro machine, I will not develop anything for other platforms.

Next project will be a racer but this is very early in development.

Fig. 3 - The game Racer, currently in development

We will probably also release the 100 level snes version of puzzle bobble at some point this year. (challenge: maybe somebody hacks the game and does it sooner ?)

N: Thank you for the interview and see you next time.

Interview with Alessandro Rogati, creator of Salagiochi 1980

by Giorgio Balestrieri and Mic the Biker

Many readers (at least the Italian ones), we are sure, are very familiar with Salagiochi 1980, the Facebook page dedicated to video games from the early days until the period '80-'90, starting from the arcades up to include those for consoles and microcomputers of the time. This project, started by the will of the creator to share his passion for video game entertainment, has collected an extraordinary success and today the page is flanked by Twitch and Youtube channels that together count tens of thousands of members. Here we propose an interview with Alessandro Rogati, the mind behind the page, who will tell us how it all happened.

GB: A few words to get to know "the man in the arcade": who is Alexander in non-virtual life?

I'm Alessandro, Alex to my friends, from Marche (Italy) born in '73, passionate about video games since a lifetime practically. I'm a simple employee in a large men's and women's clothing company and father of a fantastic 22 years old girl.

GB: What was your first contact with a video game? How old were you, where did the encounter happen, and what game was it?

The first devastating impact with video games occurred (I think) in the arcade of my small town in the hills between Ancona and Macerata, Filottrano, about 10,000 souls, in the early 80s. The old arcade (I say old because later a second one opened and we called it 'the new arcade') was next to one of the two cinemas of the village, and on Sundays coming out of the screenings we were literally enchanted by the sounds and colors that came from that dark and smelly room. The first game I saw was SPRINT by Kee Games, practically the ancestor of Atari, a game of cars in black and white made only with large pixels, but on its cabinet towered an irresistible steering wheel, gears (I think) and the accelerator pedal. I'll let you imagine the rest...

GB: What consoles and/or microcomputers have you had and which ones would you have liked to have?

My gaming 'career' is very varied and in some ways troubled (especially for my parents who had to hunt for the money to buy the various machines). It started in the early '80s with a Philips multi-game that contained Pong, Soccer, Tennis, Squash, skeet shooting and others, with an optical gun that worked a bit 'hiccup 'let's say. Then came the legendary VIC-20, followed closely by my legendary MSX PHILIPS VG8020. After crying and unspeakable scenes at home, my parents decided to buy me the king of home computers (obviously for the time): the great COMMODORE 64. In the midst of this tangle of wires and tapes, an NES also arrived, with many great games and just as many expletives. At the end of the 90s, ladies and gentlemen, he arrived, or rather she, the AMIGA 500 and after a year also the expansion that brought it to the stratospheric threshold of 1 mega. For a few months I also enjoyed a SUPER NINTENDO, lent to me by a dear friend who no longer used it, but which, unfortunately, I then had to return, otherwise my parents would have killed me. The first 'machine' purchased with the sweat of my brow was the unforgettable PLAYSTATION 1, immediately modified to take advantage of the 'thousands of thousands' CDs that were sold at the time for a small amount of money. From then on (almost) only and exclusively PC, although I also bought the PS2, but I have never deepened more than that.

GB: Favorite genre and game?

I love arcade games, especially platformers, jump and shoot and similar ones, but I played a lot of everything the market offered at the time. As arcade titles I can mention Wonder Boy, Rainbow Islands, Black Tiger, Psychic 5 and many, many more. If we talk about 8, 16, 32 bit or higher, the pages of this magazine would not be enough to list them all.

GB: In addition to video games, have you tried your hand at programming? What languages have you tried your hand at?

I'm a serial 'copier' of the first hour, when you typed the endless lists from PaperSoft to be clear, but I've always

Fig. 1 - Header of the facebook page of salagiochi1980

been a curious person, and for this reason I learned basic stealing here and there, following the Jackson courses and reading the manuals of the various machines. Only many years later, after a long break, I approached html and php and started making sites for friends and others. Some time later I also did a Flash Mx course for the same reason, which I then used to 'program' small flash games.

GB: Do you have your own collection of consoles and microcomputers?

Over the years it is hard to carry around all your belongings especially if, like me, you have gone through several removals, divorces, cohabitations and so on, but something I managed to keep. I have a C64, which unfortunately since a couple of months has started to act up and does not turn on anymore, my legendary MSX Philips VG8020, two AMIGA 500, one working and the other with a broken drive, two NES with several cartridges in tow, PS1 with a myriad of CDs, PS2 and other material. Slowly I'm trying to buy back what I would like to keep in my collection, like the Vic 20, GameBoy, Master System and more.

GB: Let's talk about SalaGiochi1980: how did the idea come about?

After a lifetime spent on video games, including also managing several sites and forums dedicated to abandonware and gaming in general, I thought I would share this great passion of mine with the rest of the world. On February 14, 2014, I created the page and wrote the first post, the rest is history. The page had an incredible boom (in 2014 we were still counting on the fingers of one hand pages and Facebook groups dedicated to retrogaming), in a short time I found myself with 1,000 members, then 5,000 and after two years we were already at 45,000. A success that literally left me speechless.

GB: Was it born as a group or did you create it from the beginning as a thematic page?

SalaGiochi1980 started directly as a page, at the time (we're talking 2014) it seemed like the best choice.

GB: To date SalaGiochi1980 has almost 90,000 "likes", how long did it take to reach this result?

SalaGiochi1980, as I said above, has been online (on FB) since February 14, 2014, so that's 8 years, 8 years of hard work, passion, joys, sorrows and you name it.

GB: When did you realize the game was starting to get serious?

My approach to life has always been 'semi-serious', and even in this adventure I did not want to change form. Things that are too serious eventually tire those who do

04:51:49
1€/100 BIT/FOLLOW +3 MIN | SUB +10 MIN
1UP 314500
HI 314500
TIME 26
GOLDEN GU
55
dallacera1971 sera a tutti
Shrapnel21 Lo 007 che non si ricorda mai nessuno è George Lazenby, comunque. =)
simobol76 Moira la tigre del ribaldabile
simobol76]ribaltabile
Shrapnel21 Al Servizio Segreto di Sua Maestà.
pippomania1927 bellissimo è uno shinobi più movimentato
matand86 00bidubi
brumbi_lambi questo boos fight è a tempo, puoi anche non sparare
Shrapnel21 !duncan_cicci
Brighel83 questa mattina ci sono.....va bene fammelo segnare
BULLETS 144
1-LIFE
ENEM
Obiettivo Abbonati Febbraio '22
23
100

Fig. 2 - Alex during one of his live on Twitch

them, but also those who watch, see, listen... I would not like it to become a job, but always and only as a fun way to spend my free time. I have a job that pays my dues already and keeps me out of the house for 8/9 hours a day.

GB: How many games did you talk about on the page?

So many. I've lost count for quite some time now, but I keep my own archive to repost from time to time, since not everyone can read the many posts we make every day.

GB: Besides Facebook, you're also on Twitch and Youtube: how do you manage to find the time to do everything and run the family? Do you have people helping you or do you manage everything by yourself?

You can find the time for your passions, often stealing it from your affections and your little spare time. The first period of SalaGiochi1980 I was alone, but then I had at my side several friends who helped me to manage everything, and I will never stop to thank them (see Tiz, Lighting, Meo...). At the moment a dear friend of mine, MRAndread, helps me in the daily publication on FB, and joins me in conducting several columns that we do live on Twitch. During the theme nights many other friends are always ready on Discord or on cam to give a valuable hand including the legendary Duncone from Malta, Andrea Pipo, EnriMassa, Marco Turrican, Simobol, MarcoVana and many others that I can not mention not to risk filling an entire A4 sheet. THANKS GUYS, from the deep of the heart. To all of you!

GB: Let's talk about costs and revenues (if you'd like and if there are any). How much does it cost to run a project like yours? Are there any forms of revenue?

As above, a passion has no price, there are people who spend $100,000 on a used Michael Jackson sock!!! Revenues? Good question. From FB and INSTA absolutely nothing. With YT and TWITCH you pay your fiber bills and a few accessories (cam, mic, pad...) per year. If people think they're going to make money by streaming retrogaming, they'd better look elsewhere. And it's definitely not my main purpose anyway.

GB: What do you appreciate in the "old" games that you notice missing in the modern ones, let's say from 2010 to today?

The games of yesteryear have heart, imagination, fantasy, straight to the point, but above all fun, an element that titles from millions of dollars, with a script worthy of Hollywood, will never be able to recreate.

GB: How do you choose the games you talk about? Do you favor one feature in particular or do you assign equal treatment to all of them?

I don't have absolute preferences, although I have my favorites ahahahahah. Let's say that I like to range a bit between all machines and genres without distinction, from the Stone Age to the PS2 (just to give me a limit), but sometimes I also crossed the line.

GB: The reborn development of games on old-timey systems: what do you think about it and who is your favorite programmer, if any.

It's great to see how rich and lively the undergrowth of indie developers is. It's something that makes our beloved machines even more valuable, and allows us to exploit them much better than thirty years ago. I always try the new titles for all the machines, it's something I really love, especially when there are remakes of great classics that can be compared with the versions developed at the time (read GnG and Commando of Nostalgia for Commodore 64... and many others). A developer? Do not make me name names... Well, let me just say one because I love his work and I also interviewed him on Twitch: Antonio Savona.

GB: If you could create a video game, what would you like it to be like?

In a word: F U N N Y!

GB: Plans for the future: where will SalaGiochi1980 go?

Good question, but unfortunately I don't have a crystal ball. What I can say is that, within my possibilities, I will try to carry on with the same passion, perseverance and fun as now.

Hi Alex, this is Mic the Biker or as someone called me "Haggar". I remember perfectly the birth of SalaGiochi1980, practically I started to publish my passion online just because of you. I wanted to ask you a few questions.

MtB: How has SalaGiochi1980 changed over the years?

Hi Michele, it's always a pleasure to chat with you. Overtime SalaGiochi1980 has undergone many changes, but the leitmotif has always remained the same: the first place is the passion for this fantastic world, everything else is in the background. Now we give much more space to live on Twitch and videos on the tube, but it is a natural evolution, to keep up with the times.

MtB: Tell us about the wonderful initiative of the video greeting for the first year of SalaGiochi1980.

Ahahahahahah, I had completely dismissed that. And in fact I don't remember much, but it's also been 7 years. I'm getting old, give it to me :-).

MtB: The first year was a whirlwind of names that, thanks to SalaGiochi1980, made friends and created collaborations. Tiz, Clax, myself, Edi, Padova and the legendary Lightning DJ. How do you remember that period?

It was a great time, we were really a team and many good friendships were born that continue to this day. We used to wait until the evening when we came home to chat with each other to exchange opinions, tips and anything else. And we didn't only talk about retrogaming! The absolute state of grace for SalaGiochi1980.

MtB: Last question: tell us about the legendary weekly leaderboards and the fantastic chaos they could create.

The famous Top Fans that was published every Saturday afternoon at 4 p.m. It was a real struggle to get the top spot because the app rewarded the people who interacted the most with the page, comments, likes, shares and more. I remember the challenges between us to the last comment in order to climb up the rankings and see our name at least in the top ten. Such good times! Thanks for reminding me of them! Hi Mic, always on top (fans) ahahahahah!

With the memory of the Top Fans ranking, our interview ends. We thank Alex for allowing us to peek behind the scenes of Salagiochi1980 and we express our appreciation for his efforts in keeping alive the passion for retro informatics that unites us, and we wish him ever greater and more important successes. For those who want to know more, in the box dedicated to the links we publish the addresses of the Facebook page, Twitch and Youtube.

Useful Links:

- https://www.facebook.com/salagiochi1980/
- https://www.twitch.tv/salagiochi1980
- https://www.youtube.com/user/salagiochi1980

Remembering Jerry Lawson

One of the great founding fathers of modern video games

by Takahiro Yoshioka and Carlo Nithaiah Del Mar Pirazzini

Gerald Anderson Jerry Lawson was an American engineer known for his remarkable work in designing the Fairchild Channel F, a console developed in 1976 that was a revolution at the time.

He is also considered the inventor of game cartridges and one of the founders of modern video games.

In 1972, shortly after the release of the legendary Pong, Jerry Lawson produced and developed one of the first arcade games, Demolition Derby. This skill eventually earned him a promotion within Fairchild Semiconductor (a division of Fairchild Camera and Instruments) in the video game arm of the company that, thanks to him, would become one of the first console manufacturers.

Jerry Lawson went on to serve as chief hardware engineer, director of engineering and marketing for the video game division, he who, as a teenage self-taught ham radio operator, earned money repairing his neighbors' televisions.

At a time when home consoles couldn't hold more games than the built in ones (the Magnavox Odyssey had cartridges, but they didn't contain memory units), Fairchild envisioned interchangeable game cartridges that could be loaded into the machines without damaging them.

From this idea will be born in November 1976 the Fairchild Channel F, simply the first programmable ROM cartridge

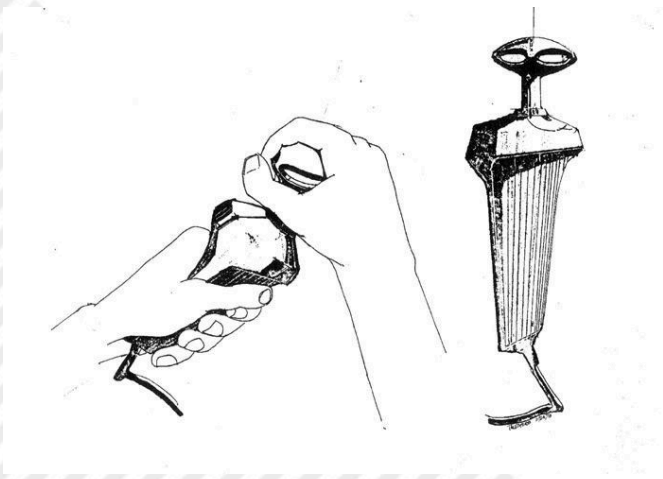


Fig. 1 - The joystick

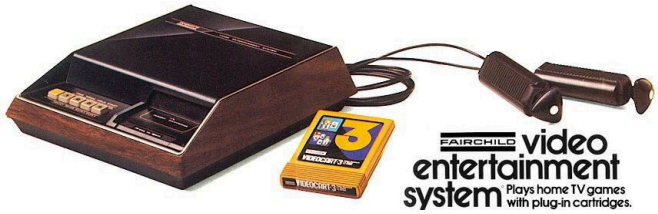


Fig. 2 - The Fairchild console

console, as well as the first console to use a microprocessor, for which Jerry Lawson also designed the prototype joystick.

It's pretty clear today that the original idea for the video game cartridge came from two men, Wallace Kirschner and Lawrence Haskel, who worked for Alpex Computer Corporation under license from Fairchild. It was then that a team consisting of Ron Smith, Nick Talesfore and Jerry Lawson perfected the technology and turned it into a commercial homebrew product. Therefore, credit for the first round must be technically shared among these five men, not just Lawson.

ATARI 2600™
COLOR BAR GENERATOR™
VIDEO SOFT
P.O. Box 4242 Santa Clara CA 95054 (408) 980-9211
The VIDEO SOFT Color Bar cartridge is designed to operate with the Atari 2600 VCS console and all machines with Atari adapters.

Fig. 3 - The cover of Color Bar Generator by Video Soft

Fig. 4 - Jerry Lawson in a recent picture

Although this system never enjoyed the heights of popularity like Atari, Nintendo, or Sega, it was a pivotal step in an entire industry because it offered for the first time cartridge circuits strong enough to withstand physical shocks and thousands of insertions without damage. The main fear was that electrostatic discharges could easily fry a semiconductor chip.

Very quickly, Jerry Lawson became a member of the Homebrew Computer Club, a sort of innovators' lobby and amateur club from which would later emerge the computer industry's greatest legends, including Apple founders Steve Jobs and Steve Wozniak. For the record, Jerry, who was visibly unimpressed with Wozniak, turned him down for a job at Fairchild...

He also founded and ran VideoSoft, a video game development company for the Atari 2600 in the early 1980s, when the console became the market leader. Unfortunately, the company only released one, a technical tool called the Color Bar Generator.

Honored as an "industry pioneer" by the International Game Developers Association in March 2011, Jerry Lawson died a month later in Santa Clara, California, at the age of 70.

His brilliant career as an innovator positions him as a symbol for all African Americans and as one of the modern fathers of the video game.

A bit of rarity

(rummoging here and there)

Warren Davis and the invention of Q*bert

by Alberto Apostolo

Many video gamers love to know the authors and the backstories related to the creation of their favorite titles.

A few weeks ago, the American publishing house Santa Monica Press LLC published "Creating Q*bert and Other Classic Video Arcade" written by Warren Davis (figs.1 and 2).

For the very few who don't know, Warren Davis is a game-designer and programmer, developer of the game "Q*bert" (fig.3).

Appeared in 1982, "Q*bert" became one of the most famous arcade games of all time, entering rightfully into popular culture with gadgets of all kinds.

Warren Davis also developed "Us Vs. Them" (a Laserdisc game) and broke new ground in game design by developing the digitization technology that was used in "Mortal Kombat", "Terminator 2 the Arcade Game", "Revolution X", "NBA Jam", "Joust 2", and "Narc". In particular, for Revolution X, he designed a kind of pseudo-3D environment on a hardware-2D.

With a brilliant and friendly style, Warren Davis tells us his story full of interesting anecdotes. It begins with his school days and how he learned to program on the Monrobot XI, a strange computer produced by Monroe Business Machines.

In 1981 he was hired at Gottlieb (where he developed "Q*bert") and a few years later he worked at Williams.

His professional career has taken

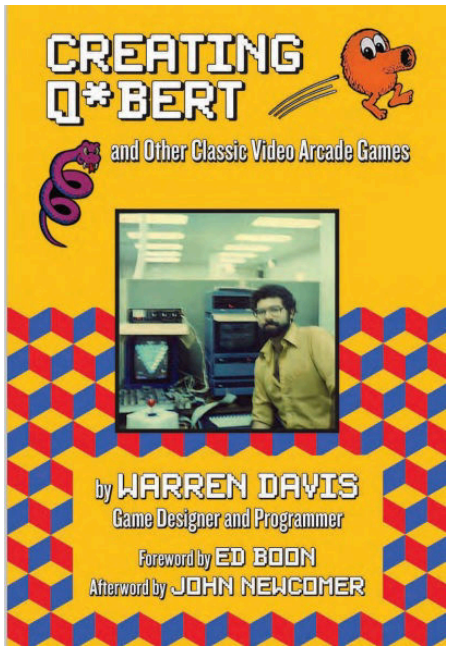


Fig.1 - Book's cover

place in contact with the most talented professionals in the history of video games: Eugene Jarvis, Tim Skelly, Ed Boon, Jeff Lee, Dave Thiel, John Newcomer, George Petro, Jack Haegar, Dennis Nordman and many others.

I think I've said enough, and I don't want to take away from the pleasure of discovery by flipping through the pages of the book.

At the moment only the English edition is on sale. It is not sure if it will also be printed in an Italian edition.

Francesco Fiorentini: I had promised Alberto to add my comment to what he has already written about this publication. Alberto, as usual, in a few lines has brilliantly described the book and its contents. There is not so much more to add. However, I would like to confirm that the book is very well

Fig.2 - Warren Davis

Fig.3 - Q*bert, the game

written and the reading flows smoothly, so, despite the fact that it is only in English, I would recommend it to everyone, since the book does not make use of complex periods and vocabulary.

Sometimes they come back...

by Carlo Nithaiah Del Mar Pirazzini

Title: Mortal Kombat Arcade Edition
Developer: Master Linkuei
Platform: Sega Megadrive
Website: https://drive.google.com/file/d/1TiBqHKlw0a3cl3f5Al4umqNxPpq7E-4_/view?usp=sharing

Mortal Kombat was a revolution! It overwhelmed us gamers with a barrage of graphics never seen before (except for Pit Fighter), simple gameplay and... a lot of ultraviolence and gore.
Needless to say, the rush to arcade conversions was devastating. Every system of the time had a more or less decent conversion of this Midway's game.
The versions that stood out the most at the time were those for the Amiga, for MS DOS and those for 16-bit consoles.
While the Super Nintendo version was heavily censored by Nintendo itself (solvable with a trick), the Megadrive version, although technically inferior, was a great game.
After a long time the developer Master Linkuei, famous for his way of "fixing" the console arcade versions of many titles, has decided to release this hack that puts in order the game bringing it to arcade perfect, or almost.

What has been added? We have a better graphical cleanup and re-editing of the coloring, the animations are smoother thanks to a code that takes better advantage of the game machine. The sound effects of the arcade version have been totally added and all the tips and secret modes not present in the original version for Sega Megadrive have been enabled.

Needless to say, the game still remains incredible to play even after all this time.
This hack rom is freely downloadable, works great in emulation and on real hardware.
Highly recommended.

R-TYPE - Shoot'em up year zero

by Mic the Biker Novarina

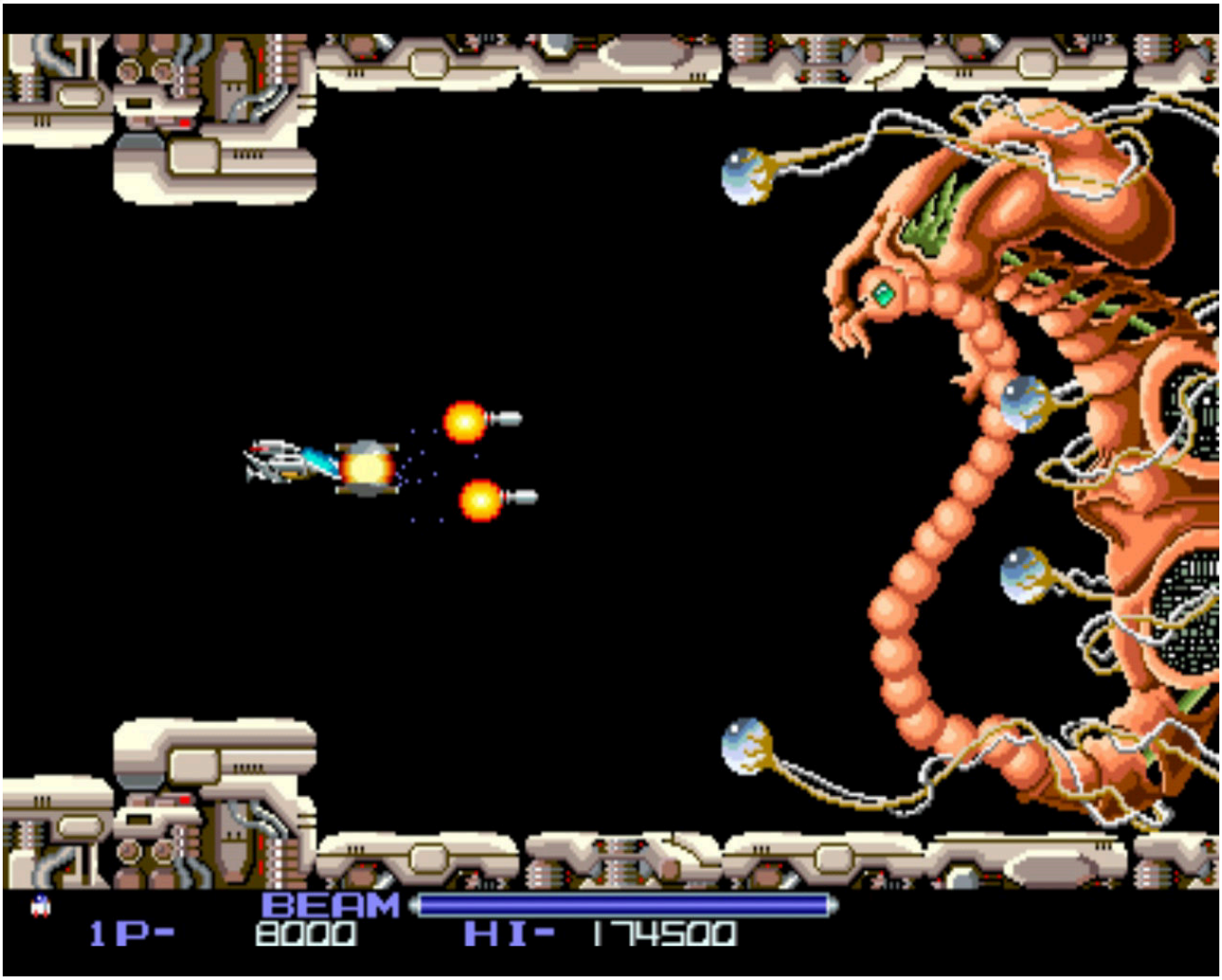
Do you see it parked there? Right next to it. The legendary DeLorean from Back to the Future is ready to take us to the end of the Eighties, precisely in 1987. In that year, R-Type arcade machine arrived and the horizontal scrolling shooter will not be never the same. A real revolution in the history of shoot em ups. In fact, this monster game follows an infallible market rule: it succeeds once in a lifetime. Let's start with our travel friends, with our coin in our pocket. Entering the arcade room and seeing the cabinet, the genius of H. R. Giger, aka the creator of Alien and the greatest exponent of organic biomech style, cannot fail to come to mind. However, what the hell is this game? Mysterious object branded Iren, it has managed to arrive with its charm intact to the present day, as well as its immortal R-9 battleship that made school in the genre's design. A strictly horizontal shoot em up with an arsenal that can be upgraded with four different types of weapons, out-of-scale graphics in the true sense of the word and fantastic sound. Until that day, nothing had gone that far: there was a row behind the cabin even with no one to play with it! However, apart from the unseen graphics and the mad sound, was there really anything else? Absolutely yes. It was since the dawn of the first shooters that no major novelties were introduced. First of all we have the Force Device, the famous sphere that doubles the firepower, that become a must from that day and. We can throw it against opponents or control and attack in front of or behind the R-9, so you can fire in both directions! What's more, the Force Device absorbs the less powerful shots of the opponents, becoming a real shield. A brilliant idea, united with a dynamic that allows it to be exploited in an intelligent way, reducing that spasmodic need to shoot faster and faster. Thanks to this mix, the fun is elevated: R-Type is incredibly fun! Unlike its predecessors, it is not enough to memorize the enemies' paths: here you have to use the disposable resources in an appropriate way to the level in which we are or looking to the series of enemies coming. If the game had already stopped here it would have been enough to annihilate his competitors but…

Beam is coming!

As if that were not enough, R-Type also adds another element: the Beam! Hold down the fire button and you will see an energy bar charge up. Once at maximum, the R-9 will release a devastating beam capable of resolving situations at the edge of the human. Think about how revolutionary was: you had to think in a matter of seconds the real benefit due to the time lost to load the weapon and the real damage caused to the incredible enemies. Here, I wanted to stop right here. Never before enemies seems escaped from a nightmare: the first huge monster, the biomechanical heart as big as the whole screen, is like a nightmare we dreamed. For the first time the enemies, the monsters, are a backbone of a game, fundamental for the genre evolution. Malignant biomechanical creations, chilling, ancestral, original and very, very large. Terrifyingly big. The final guardians show off mercilessly annihilating full screen dimensions! The R-9, which until a few moments before seemed immense, becomes in no time a grain of sand in front of the majesty of the xenomorphs, just to pay another tribute to Giger.

Bosses have a weakness

Giants with feet of clay, because here too the dynamics



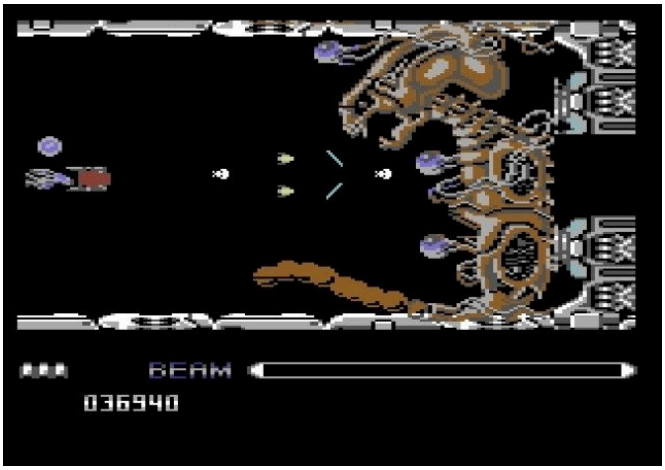
are different: they go down with a few shots, but must be fired at a tiny weak point. A contest of nerves, because the giants in question shoot. How furious we have been, how many tokens we have thrown into this hallucinating coin op, if only to see what the graphics were like later on!

Of course, like all shooters, it's not a perfect game, the genre itself can't be. However, the shoot em up is the essence of arcade rooms! The difficulty of the game is well calibrated. We assume that, when you become skilled, you can finish it after a few games without ever dying. Problems arise when you are killed and have to restart from a previous point too far and, above all, without armament! In this condition, it is impossible to win. But we didn't give up: another coin, we have to pass that damned level. We have all played this game at least once; we have all participated in our own way to make it an immortal masterpiece and a crossroad for the genre. Nothing was the same after him. In 1987 its impact was like the monolith of “2001 a space odyssey”: it elevated the genre. R-Type extinguished the old concept of shoot'em up, changed it, made it evolve and made it better. I still remember the frenzy of playing it, of rotating with friends to pass the levels. In addition, I remember how anxiously we were waiting for news about possible conversions for our home computers or consoles.

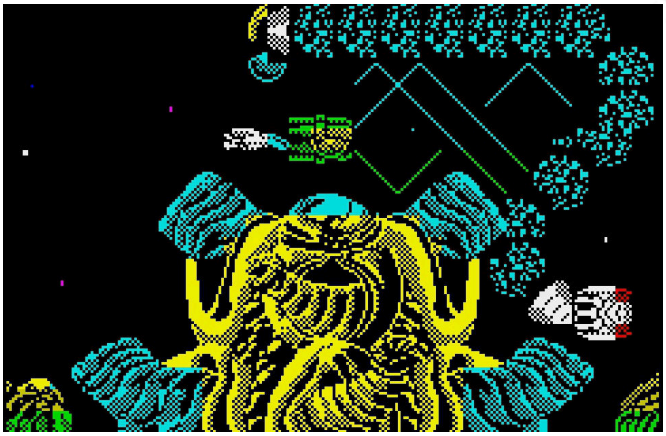
And arrived Katakis

Every Zzap! issue was probed line by line in search of confirmations which, given the galactic success of the game, were not long in coming. News of a conversion for the Commodore 64, my home computer at the time, began to leak. Activision announced the making of the game with great emphasis. However, while we were waiting, on Zzap! Number appears a review of a Rainbow Arts game called Katakis.

Thunder and lightning, this is R-Type with a different name! A legal dispute arose and the title was withdrawn from the market. In the meantime, we had all some



Rtype on Commodore 64



Rtype on ZX Spectrum

pirated copies of it, and it was a great game. Activision saw the prowess of Factor 5 team, which developed the game, and hired them for the 1989 Amiga conversion. Therefore, shortly after, Rainbow Arts was able to release Katakis , changing its name in Denaris. Soap operas were in vogue at the time, even in our industry!

Home conversions

Let's talk about the best home conversions of R-type. For the good C64 the conversion, considering the machine limitations, was satisfactory but perhaps, considering the title scope, a little overestimated. Let's start with the porting, leaving out the graphics: some enemies do not perfectly follow the attack trajectories that were present in the coin-op. In my opinion, this aspect definitely cuts the conversion, because much of the gameplay is based on the types of attacks brought by the enemies. Removing this factor reduces everything to a trivial shooter. The controls lack the second button present on the arcade, and having to press the space bar to use the Force Device is not the best. The same, when detached from R-9 ship, make useless trajectories.

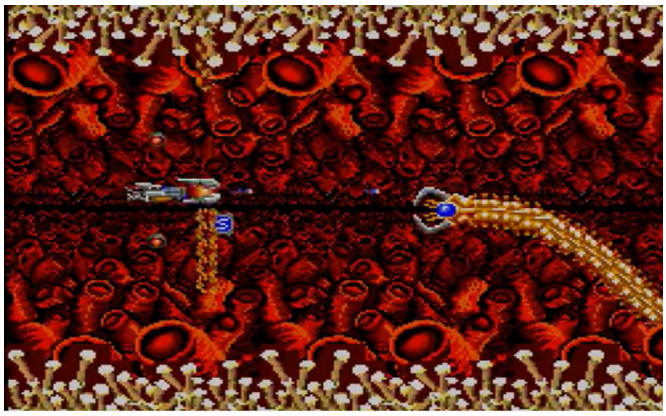
Once the defects are over, let's go to see the merits. The graphics are generally of good quality and are very similar to the original version, including a basic parallax for the background. The sound is stratospheric: the good SID is a guarantee and it brings out a perfect soundtrack. The conversion maintain that biomech atmosphere that made the original famous.

The conversion that struck me most was the one for the glorious Zx Spectrum. Despite having its known hardware limitations compared to its competitors, the programmers managed to make the graphics sector splendid. The good processor speed allowed for smooth and precise gameplay. The final rendering was indeed very colorful, with limited block effect and gorgeous detail. A dear friend of mine had the Spectrum and, although we used to stick between c64 and spectrum users, this time I made a standing ovation.

Staying on the home computer, let's move on to the Amiga version, the 16-bit machine from Commodore. As mentioned before, the conversion dates back to 1989 thanks to Factor 5. These Teutonic programmers did a good job that brought them to light, even if the conversion has major gaps. R-Type for Amiga is not a flawless product: the fidelity of the conversion is lost even having an excellent programming job. The big flaws here are the oversights compared to the original, and it may happens, but also if compared to other transpositions for less performing systems: this cannot be forgiven. Graphic deficiencies that will leave a bitter taste, considering Amiga characteristics: we could have an almost perfect

Rtype on Amiga

Rtype on PC Engine



Rtype on Master System

conversion. The Amiga version of R-Type lacks any type of parallax present in the original version. An incredible lack since Amiga was able to handle parallax like few others, just see Shadow of the Beast. The graphic effect of beam loading has disappeared. Many backgrounds were removed, for example in the third level the ones that surrounds the enemy spaceship. This has fewer parts to destroy, it has been made too easy. All game play was simplified, too much to be honest. As written at the beginning, the coin op is certainly not impossible, but it requires good experience and meticulous mastery with controls. The Amiga version, also compared to the 16-bit competitor Atari ST (of which you can find a detailed review in the Italian Retromagazine number 5), it drastically halves its content. I will give another example: the second level biomech worm is shorter even than the Atari ST version. Perhaps I think it was inexperience of the programming team and a relative youth of the machine: anyone can notice the differences from the original.

I left for last the two best R-Type conversions, because they are truly amazing. The first is the one for PC Engine. This was a historical combo, a certainly casual combination: thanks to this conversion, the PC Engine was known throughout the world. The console was very popular in Japan but it was almost a mysterious object in the rest of the world. Here we have only the first four levels of the coin op, but the care and fidelity to the original were absolute. It was a winning bet: one of the most popular arcades of the period converted on domestic equipment with almost no graphic or sound loss. There was a time when you said R-type you think PC Engine and other way around. Unfortunately, very few machines with the name Turbografx arrived in Europe, and it remained a dream for almost everyone. A dream come true only in our days with the release of the Pc Engine Mini version, in my opinion the only old console replica that really makes sense.

Finally, let's take a look at what Sega realized for its 8-bit console. For Master System this transposition is incredible: even if the sprites present on the screen are slightly fewer in number, all the scenarios are very close to the original version! The graphics are so sharp and colorful that looking at the game it seems to be one of the first Mega Drive games. The duty to pay for a similar graphics is some flickering and slowdowns. The sound is even more impressive than the graphics, for its similarity to the coin op, remembering that the Sega eight bit was not particularly equipped with dedicated chips. Concerning the playability, we are perfect and implements it with such professionalism that we do not regret the original bar. As we have seen before, the flow of enemies was one of the strong points of the coin op and here they are practically the same, both for the attack patterns than their position on the screen. Do we want to find it a bug? The game rhythm is slightly slower than the original one. However, Sega guys wanted to do more: they added an entire level with its secret boss! Can I tell you how to access? No come on, it’s better you will play the game, you won't regret it.

C’mon, it’s time to be children again. Blast off and strike the evil Bydo Empire!

The Age of Heroes

Mic the Biker Novarina

We don't need the DeLorean today, because we are going to know a new one for our beloved Commodore 64. It is now well established that the retrogaming world is no longer just a dusty room for old nostalgics. What was initially viewed with sympathy as a movement of elders who evoked old glories, today has exploded into something new. A wave with the strength of a Tsunami made up of new and fantastic games, which is ridden by the mighty old 8 and 16 bit platforms, now in better shape than ever. Here, The Age of Heroes is part of this new videogame stream.

While the new consoles limp, thanks to a not so bright historical period and an electronic components crisis never seen before, the old gen gets new and shows a great physique. The new entertainment systems are almost at a standstill, while the old glories gain more and more space in the hearts of fans, even under 40. Of course, I'm a fan, I grew up with these systems and I love them but objectively they have several advantages. Taking the good Commodore 64 for example: never received a system update in 40 years. His games? Pick them up, load them up and play. You can do it on the original machines of the 80s, on the reissues in a mini or maxi format or you can emulate them almost anywhere. And the same goes for the remaining old glories of the past. In contrast to this, I find it stressful to manage next gen consoles, from Play 4 and Xbox one onwards. You turn them on and there is already an update of a few giga waiting for you. Ok, but what is there to update continuously? After that, I inserted the game disk, because I got the physical edition in store, which cost me less than the digital one. Magically, not even the time to see an image and on the screen appears that warns of a 30 giga update to start the game. What did they put on the disk then? And it is continuous, between DLC, other updates, upgrades. Then the game after a week broke me and I threw it in a corner.

Maybe I understand...

I begin to understand the situation! If I have half an hour and I want to relax for a moment, retrogaming exactly does what it was created for more than 40 years ago: have fun! Everyone noticed that, and the development of new games is now no longer a rare fact but a solid reality. Today we see games for our old electronic gadgets that if they had come out in their day would have given us a lot of fun, and in some cases, we would have screamed at a miracle. Therefore, here we are to see this funny Hack n Slash that takes RASTAN as inspiration. Psytronik is a guarantee: it has always accustomed us to good games with great dynamics. The Age of Heroes is no exception: it was released a couple of years ago but, being part of this "back to the future", perhaps not many will know. The team that took care of the game is a top one: programmed by Achim Volkers, graphics by Trevor 'Smila' Storey, while the sound

is a guarantee thanks to Saul Cross. The story tells us that an evil specter is sending his legion of the damned across the frozen wastes, mountains and sacred temples. It will be us, fearless protagonists with a Joystick in hand, who will have to remedy this mess, traveling and fighting to eliminate the ghost and the bad legions. After all, we are the descendants of the warriors of light: can we refrain from fighting?

The game

After a great loading screen, and a selection between a male or female warrior, we begin our journey. Here we are, struggling with a side scrolling arcade, armed only with a sword and a spell. As in the best tradition of the genre, enemies will attack in waves from both sides, trying to make life difficult for us. The graphics are very good: they use the black color very well to create high contrasts and satisfying backdrops. The sprites are well animated and the sound is great. The feeling is nice: it's practically like going back in time. Obviously, the game's dynamic is that, it cannot be otherwise. I appreciate the willingness to insert some variations, such as in the second level: here we will find a boss to eliminate. Very nice, both for the atmosphere and to give more depth to the game, the idea of the paper map that shows us the levels that are blocked or not.

My joy is always at the top when I face these old school gems. What surprised me positively is the initial difficulty, quite easy: you get to the first boss without great problems. Of course, the dynamics are always the same: you go on, you jump, you use the ropes to go up and down and you hit enemies with your sword. If you are thinking "Like in Rastan!", you are right. I believe that the developers took the famous arcade as an example to follow, making the initial approach easier. This is due to some feedback from players in beta testing. Therefore, the development team decided to unlock the hardest level only after completing the game.

One aspect that I liked is that in the game you can level up. Like any good self-respecting RPG, killing enemies accumulate points, which will make us advance and become stronger. The visual effect that accompanies this increase is great, and is perhaps the best thing about this aspect of the game. When you level up, we recover all the energy, and also will add a little more. I was disappointed with the way points are stacked to advance: only melee kills against enemies give points. If we use magic, nothing happens, much less killing a boss. There are some bugs in the game, or at least I thought they were. It happens that I land slightly too high to hit enemies, I couldn't hit them even getting down. Sometimes the bad guys walked through me without hurting me, which isn't so bad for me! Nothing so serious as to affect the playability, after all, even a sacred monster like Rastan for C64 had some problems. The game is available in different formats: they range from the standard cassette, to a special edition on clamshell tape (limited to 50 copies and I think already finished). We then have two versions on disc, one inexpensive and one premium. The top remains the Collector's Edition box set with the disc game, soundtrack CD, poster, keychain, badge and stickers! I'll post the link for details on how to order physical versions as well as download from csdb.

https://psytronik.bigcartel.com/product/the-age-of-heroes-c64

https://csdb.dk/release/?id=176388

FINAL JUDGEMENT

GAMEPLAY: 90% You can't think about Rastan while playing it, but luckily the game is not limited to that. The action is not lacking and it is well calibrated: slashing is always a great pleasure.

LONGEVITY: 75% Like most of this genre of games, once finished it will rarely be reloaded. Too bad, since only by finishing it, the difficulty initially conceived by the programmers is activated, a real challenge to try again!

NEW GAME

THE SHADOWS OF SERGOTH

Year: 2020
Publisher: Double Sided Games
Genre: RPG
Platform: Amiga
Website: https://doublesidedgames.com/shop/commodore/commodore-amiga/the-shadows-of-sergoth/

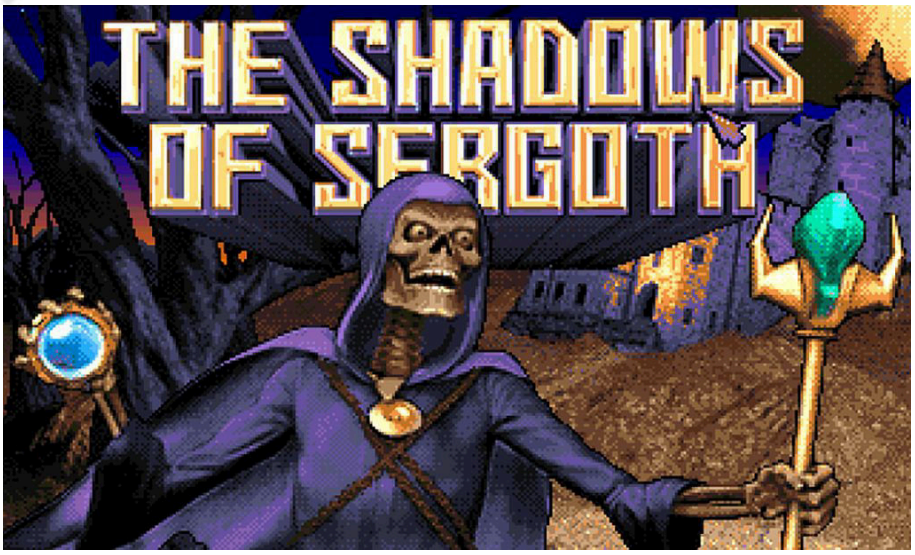
A good old-fashioned dungeon crawler on the Amiga? How wonderful! I loved the Eye of The Beholder series and this The Shadow of Sergoth is an heir to it.

The game takes place in the medieval fantasy world of Chrisandia, on the Marak Peninsula.
A not very populous area where the population makes a living from trade and fishing.
Its current king, Orlof VI the Valiant, is a member of a long dynasty of enlightened rulers. During the wars against the evil emperor Sul Rakin, King Orlof showed great valor and the kingdom gained a victory over evil but many losses.

The events in the game take place five years after the fall of Sul Rakin, in the year 351 of the Crystal Age.
It is a classic underground exploration role-playing game reminiscent of the aforementioned Eye of the Beholder or Dungeon Master. It is based on Microlite20, a simplification of the Advanced Dungeon & Dragons rules, and it is born based on the version of the game for Amstrad CPC6128 that came out in 2018.

The game requires an Amiga with at least 2 Mega Ram chip even if I personally recommend a faster Amiga (Amiga 1200 or accelerated) to activate the 3D movement effects and improve the fluidity of action.

The game however runs on any graphics chipset (OCS, ECS or AGA) and also runs configured in WinUAE emulation and on PiMIGA.
The character, as in the most classic



RPG, can increase experience and then level up to the twentieth. Obviously it will be possible to configure your character in the most classic way and level up just like in AD&D.
There are 5 very special races, in addition to humans, dwarves and elves we can also play the half-orc and lizardman. This allows a party very varied and particular (the lizardman is wonderful).

The mapping system is fortunately automatic and this thing will really come in handy compared to mapping like in the old days.
There are 20 very large dungeons to explore and there is a good save system. It is also possible to install the game on a hard disk (recommended).
The box contains three 3"1/2 diskettes, the English manual, a real compass that you can use in dungeons (crazy but true, ndN), a collectible D20, several stickers and the link to the ADF download.

Having described everything to you, let's move on to the evaluation.
Is it a worthy heir to the SSI or Dungeon Master series? Not entirely, or at least it's a beautiful game, but it doesn't quite reach the perfection of the storytelling of the aforementioned games.

The adventure is playable and linear and has a good degree of challenge but the dungeons are "simple".
Graphically it's a gem and I must admit that they put a lot of effort into making it look good.

The same goes for the sound, atmospheric and functional.
But.... But it lacks that extra something to make it an incredible game and an absolute gem.

It's a good game that deserves a respectable grade, but on the step the EOB saga remains stable.

by **Carlo N. Del Mar Pirazzini**

OUR FINAL SCORE

» Gameplay 80%
Classic but fun game system. It's a dungeon crawler and going after monsters is fun. Nice spells present.

» Longevity 75%
The game is nice to be played but the dungeons are not challenging, you risk ending the adventure too soon. Funny the creation of the party.

NEW GAME

PUZZLE BOBBLE
(COMMODORE 64)

Year: 2022
Developer: Bas Scheijde (aka Acied)
Genre: Puzzle Game
Platform: Commodore 64
Website: https://csdb.dk/release/?id=213965

I'm in exile because of COVID at uncle Nithaiah's home. He is very excited these days! Puzzle Bobble for Commodore 64 came out and he's really excited about it.
He hands me the Commodore 64, tells me how to make it work and says "Ingrid, I have to write a few lines about this title, you have been a player since you were as big as a dragon... tell me what you think?".
I couldn't refuse...and I was right!!!

Puzzle Bobble is an arcade puzzle game made by TAITO in 1994 and then converted for some consoles later. The game brings back to commercial success the two cute little dragons of the video game Bubble Bobble, also by Taito.

The player must use a cannon to shoot causal colored balls (bubbles) into a series of colored balls scattered around the game screen. These fired balls stick to the ones present, and if you manage to form a group of three or more equally colored elements, they dissolve. You have about 5 seconds for each throw. If this does not happen, the ball will be fired automatically.
The goal is obviously to eliminate all the colored balls in the level within a certain time limit and continue. From time to time the playing field shrinks in the lowest direction (the ceiling lowers). If one of the balls in the field reaches the lower part, i.e. the area where the cannon, Bub and Bob are located, the game is over.

Speed and accuracy. Essential to pass the level.
No official commercial version was released for the Commodore 64. In

1994 Commodore's 8-bit machine was already on its way to sunset.

The project was long and ambitious. Bringing back to an 8-bit machine the look of the game as faithfully as possible seemed impossible and instead.

It's all beautiful and has everything us Puzzle Bobble fanatics are looking for! Everything works great. I loved the graphics, colorful and animated really great. There's even a home screen with Neo Geo written on it.

Graphically a jewel indeed. Unthinkable for such an "old and limited" machine. Even the sound seemed appropriate and pleasant, perhaps slightly underwhelming compared to the graphics, but perfect in context.

The game supports both joystick, pad and paddle. There is a fun two-player tournament mode that is done with the corresponding number key:

1. Random (random levels)
2. Random Endless (infinite random levels)
3. Levels (levels in order of play)
4. Infinite levels (levels in order but infinite).

The two-player version gives this title eternity!

The author in the initial loading has included a trainer that allows you to skip some levels.

What more can you say? It's Puzzle Bobble, one of the most playable titles ever made. Fun in single player, never predictable difficulty level and practically eternal.

Download it and support the new developer scene.

Fun fact. This version uses some loading screens where the future projects of this incredible development team are presented.

I almost forgot... Don't miss the interview with Bas Scheijde in this same issue.

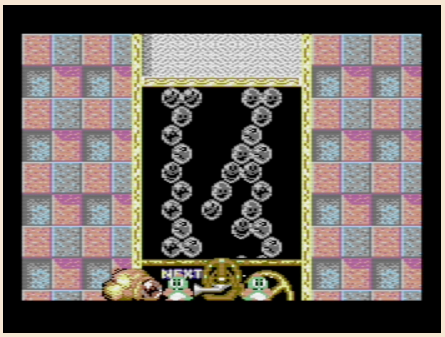
I'm going back to the game.

by **Ingrid Poggiali**

OUR FINAL SCORE

» Gameplay 99%
Simple, balanced and fun in all its game options.

» Longevity 95%
32 levels in single is a nice challenge... but in double and with the different game options it becomes eternal.



NEW GAME

WIZARD OF THE BOARD

Year: 2021
Developer Daniel Savage
Genre: RPG/Puzzle/Action
Platform: Nintendo 64
Website: https://mega.nz/file/h35G0A5b#zOA2MQtH_BwOaU_btPDiceXnudm60APAqvvJMKCSYfA

Wizard of the Board is a Nintendo 64 homebrew game made by Daniel Savage and his team and released on December 7, 2021 for the 64bre Game Jam, a competition of Nintendo 64 developers and fans.

In the game we will play a young warrior named Zeff who is forced to face demons, monsters and wizards inside a tower and will do so with the use of a... Chess!

The game won the Game Jam for originality, graphic appearance and innovative gameplay.

This is one of the few homebrew games with an accompanying story, and that surprises us. Often in competitions everything is done as straightforward as possible without spending time to create the "background" of the title. The story is explained through several introductory screens and, as we said, tells us about the exploits of young Zeff who is doing everything to become a valiant warrior.

The last step in his training is to delve into the Tower and take part in numerous tests before facing the Council Wizard.

The game is made up of all kinds of encounters, just think that the first character we'll meet in the tower is a mouse looking for cheese, quite friendly.

Often during the game we have flashbacks of the protagonist and his training.

The demonic enemies that inhabit the tower are talkative and communicate to our poor protagonist the terrible tortures they will do to him.

The gameplay of Wizard of the Board is not simple at first glance, but it makes sense and becomes addicting once you get the hang of it. Our hero's powers move giant chess pieces along a chessboard (the floor) according to the same movement patterns they would in a normal board chess game. The goal is to place all the pieces on the target spaces and, in later stages, defeat the enemies on the screen as well.

We are able to do this by selecting a piece, looking at it and pressing the pad's A button, then selecting a target destination with the C buttons or a nearby space with the Z button. This can seem very confusing because of the first-person perspective, but with a little practice the game becomes fun and addictive.

There are five types of demons that we'll have to fight against. Froggers, large frogs that jump from left to right and vice versa. Annoying but predictable.

Blade Traps that move in a straight line until they reach an obstacle, after which they change direction. They follow a fixed path until we place our piece, at which point they change

gear. Hostile.

The Smiley Spitters, strange yellow monsters that shoot projectiles, and so we will be forced to move fast to avoid them. The bullets are slow but definitely lethal.

The Blue Demons, huge demons that will chase you relentlessly if you enter their field of vision. You have to hide and plan. They must be knocked out with three shots.

Demon Queen floats on the board staying away from the player but shooting projectiles. Hostile.

The control system seems shaky at first, but once you understand the controls everything becomes a blast.

The character moves with the D-pad or with the joystick lever while with the C pad we move the cursor. Frenetic in case of enemy assault, but fun.

Despite being a completely different genre, Wizard of the Board reminds us of another Daniel Savage game from 2020, Lunar Assault 64 (a nice shooter title). It has distinct levels and intermediate dialogues that explain the story, the setting is reminiscent of the lunar world already seen in the shooter and even the enemies resemble space monsters. Who knows maybe the creator of the game will make a related saga. We'll see.

Combining chess, puzzle games and FPS is ambitious but definitely unique. Elaborate and complex to get familiar with the game mechanics, but fun.

It is not punitive. The confrontations with the demons will only come towards the middle of our journey, after we already understood what to do. A plus point. It's rare to find homebrews that guide you perfectly through the mechanics.

A N64 controller is recommended if you play in emulation to have the correct key mapping available. The game has been released in PAL and NTSC versions and works perfectly on real hardware via Everdrive.

The puzzles are not complicated to finish but offer a balanced difficulty and, once completed, a nice feeling of victory.

Conclusion.

Wizard of the Board is a great gaming sensation. It feels like we're back in the golden age of N64 titles especially with the design style. I liked the progression of the levels with the short interlude movies (which according to the creator are inspired by Puyo Puyo), they help make the game more than just a series of levels.

It has a decent length for a homebrew game and definitely looks more complex than some recent or past N64 titles.

The only down note is on the audio compartment; it is not exactly the best one… but it's a small note.

Overall a very good game that can be completed in about 1 hour of play.

Well-deserved win at the 64BREW Game Jam 2021 competition.

by **Carlo N. Del Mar Pirazzini**

OUR FINAL SCORE

» Gameplay 80%

A mix of fps, puzzle, chess and action game. Well structured levels.
Challenging control system at first.

» Longevity 90%

Decidedly well-balanced difficulty of the game and its duration.

N64brew GAME JAM

NEW GAME

SILVER VALLEY

Year: 2017 digitale - 2022 edizione fisica
Developer: Enriquez Ruiz
Genre: Action/Metroidvania
Platform: Sega Master System
Website: https://www.smspower.org/Homebrew/SilverValley-SMS

The homebrew scene of the Sega Master System, the Sega 8-bit console, is very alive and active and over the years has given us some very unique and captivating titles.

One of them is Silver Valley, an action that we will define in a modern Metroidvania key.

Actually, the title presents a variety of different elements that don't categorize it in a specific genre, but for convenience and especially for vision it recalls classics such as Castelvania or Metroid.

The story is very classic. Freeing our protagonist's world from evil. End.

The development style of the game is a combination of the Konami and Capcom classics of the time and some typically British titles that we often saw on the Amiga world.

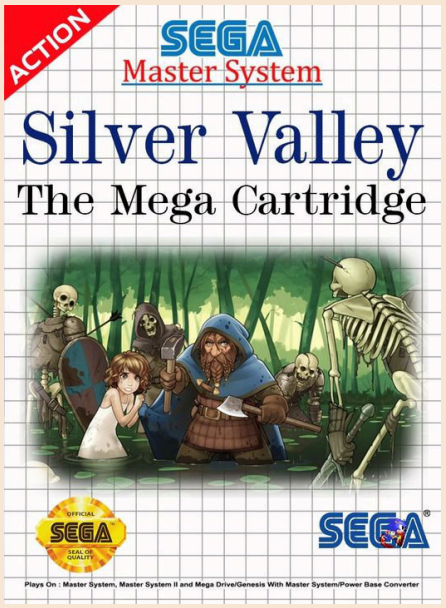
Responsive controls that, with two keys, allow us to jump, attack opponents, cling to various objects, crawl and so on.

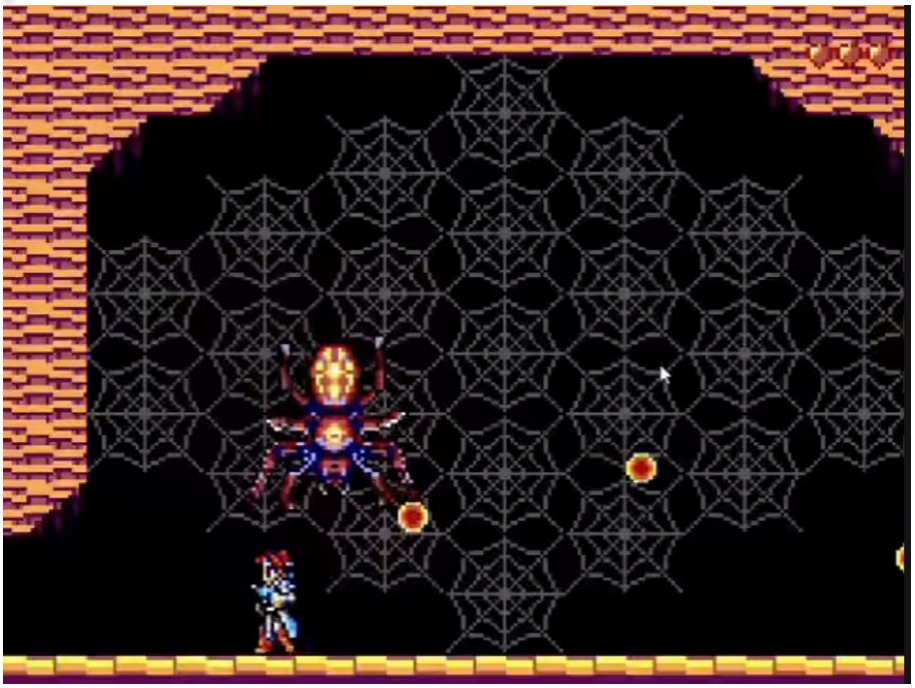
Technically and visually it is perhaps one of the most beautiful and varied titles for the 8-bit console, at least of the new era of homebrew.

Some levels are colorful, such as the shoot em up style one, others are rich in animations. The effort of the developers was huge to make everything run very well.

There are a couple of minor issues from a design standpoint, though. Enemies, for starters, almost all seem to be "damage sponges" aka leathery and require several hits to kill. This happens right away from the first game world and, in some cases, can be daunting and hostile.

It's easy to die or get hit repeatedly without killing the monster in front of us. Sure, infinite continues are present but it gets frustrating. I would have preferred to see a different system of increasing the power of weapons and less possibility to continue. It would have given me the appearance of a more "complete" game.
But I don't want to be hard on Silver Valley, because it's obviously a project born out of passion by its creator and it's also an admirable effort.





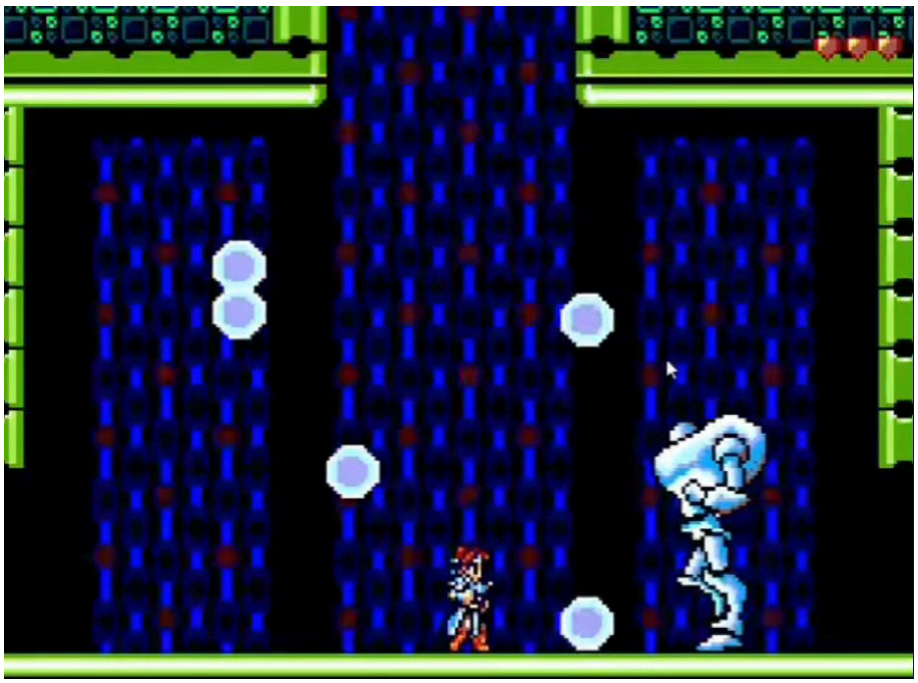
OUR FINAL SCORE

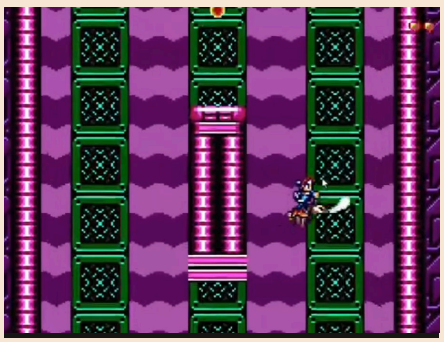
» Gameplay 90%

The control system is complete and allows us to do whatever we want. Excellent characterization of some levels and the variety of the game.

» Longevity 70%

It could have been done better in terms of difficulty and definitely lacks a more solid "structure" for weapons and damages to inflict on enemies.





If you want, the free version is available to be downloaded from the site that I linked at the top of the description of the game, but since a few days ago, there is available the physical version on cartridge that has a very well done artwork and a great multilingual manual. If you have the physical console the purchase of this version is worth it.

In conclusion Silver Valley is an impressive technical product on the small 8-bit console with some design issues and an unbalanced difficulty.

by **Carlo N. Del Mar Pirazzini**

NEW GAME

KNIGHTMARE

Year: 2022
Editor/Developer: Konami MSX e HOFFMAN AMIGA
Genre: Shoot em up
Platform: Amiga
Website: https://hoffman.home.blog/

Konami is known for producing great quality games. There is no doubt about that. Most of us long-time gamers have played the titles of this great company with great enjoyment. Gradius, TwinBee, Silent Hill, Metal Gear Solid and Castelvania are some titles that need no introduction.

The Japanese company has been around for a long time in the industry. They already got a name in the early 80s with titles like Frogger or Scramble in the arcade sector.

In 1986 they released a title called Knightmare for the MSX platform. A platform much loved by Konami and supported with hundreds of fun titles. When it came out it was an advanced game for its time. It was a great success, so much so that the development house went ahead with a second title and a third.

Today, thanks to Hoffman (who has recently converted the first Metal Gear for Amiga that we reviewed in RMW issue 31 ITA and 09 ENG) we find ourselves in front of the Amiga version.

The background of the game is quite simple. We will control a knight named Popolon who is on a mission to save the Goddess of Love and Beauty, Aphrodite.

The poor goddess is held prisoner by the wicked Hudnos, evil prince of darkness.

Knightmare is an upward scrolling shooter. As in other titles of the genre we'll have to survive the continuous assault of dangerous enemies. They come in all possible forms: bats, skeletons, black slimes, orcs, dragons. To confront them we'll have at our disposal an arsenal of respectable weapons that we'll be able to find throughout the game. Each weapon has pros and cons, so it is necessary to have the right weapon at the right time. There are also power-ups that we can collect during our adventure. Protective shields, invincibility potions, timer locks and much more.

At the end of the level, an evil end-of-level boss will show up and it can be defeated only by finding the right "system".
Knightmare is a difficult game even in this incarnation on Amiga, as it was on MSX. It requires practice and patience and a good memory.
How is this Amiga version different from the MSX version?

The graphics have been remastered and made more modern and enjoyable by Toni Galvez. Carefully animated and very varied.
There are new loading screens, you can play the game at 50hz or 60hz and some goodies have been added, like a new secret bonus.
I was very impressed with the audio department. The music and sound effects are really beautiful. A masterful job.

The game can be freely downloaded from the site that I put in description.

It is in ADF version so it is playable on emulator, gotek and you can also load it on floppy disk and play it on real hardware.

It works on all Amigas with 512kb of Ram + another 512kb. On A500 it works like a charm.

An installable version is also planned for the coming months.
Inside the zipper file you can also find the artwork that the author has made.

In conclusion, it is a very playable game even in this "modern" version. It was on MSX and the gameplay and fun remains solid even on Amiga.

As I said it's difficult, but enjoyable and lets you play it over and over again before finishing it.
Download it and support the author, he is making great products.

by **Carlo N. Del Mar Pirazzini**

OUR FINAL SCORE

» Gameplay 90%
Fun and well structured. A must on MSX. A must on Amiga.

» Longevity 90%
It's difficult but it keeps you glued to the joystick like few others.

NEW GAME

RETALIATE C.E.

Year: 2021
Developer: Ivcabral
Music: Richard
Editor: Ivcabral
Genre: Shoot'em up
Platform: Commodore 64

It is now well known that the video game scene for the eight bits (and even the sixteen ones) has returned to shine as in the past. New games, conversions and improved reissues of old masterpieces are coming out at increasingly shorter intervals. In this one I find myself playing and reviewing RETALIATE C.E., a game originally released for the Roku platform, a streaming box/TV OS.

There were already versions for Commodore 64 of this shoot 'em up outside the lines: the first versions are dated 2018. In 2019 saw the light the DX version, of which we will find a link to purchase in this C.E. (Community Edition). Originally released on December 30, 2021, it is reported that the following day the legendary guys from Army of Darkness released the cracked version, where we will find the ability to make infinite energy, missiles, and phasers.

So what does this excellent video game consist of? We have in our hands a shoot'em up of an excellent level and above all unique in its genre. Unlike the classic stereotypes of this genre, in Retaliate you start the game without bullets.

I will not deny my surprise, having never played the previous releases, in pressing the fire button and seeing nothing come out of the ship.

The joystick was well connected, everything moved but there were no bullets on the screen. I was dying in repetition, until the moment when the experience of videogamer led me first to press various keys on the Commodore 64 and, later, to try to hold down the fire button, almost looking for a weapon like the beam

~ PRESENTED BY ~
ARMY OF DARKNESS
FOR ALL OF THE LATEST C64 WAREZ... UNT
RETALIATE CE V3
12/31/2021
~ MEMBERS ~
BLACK BEARD PEACEMAKER BOARD RIDER
ONE EYED PIRATE PROWLER OPTIC FREEZE
THEFATMAN PLUS ~1~

SCORE:000000 Hi:000000
RETALIATE
YOU ARE A VALIANT SPACE HERO THAT UNFORTUNATELY IS OUT OF AMMUNITION.
NOW YOU MUST ACTIVATE YOUR SHIELD PRESSING YOUR JOYSTICK DOWN TO KILL ALIENS AND ALSO ABSORB THEIR AMMO THEN HIT FIRE AND RETALIATE!
BUT BE CAREFUL, YOUR SHIELD ENERGY ONLY LASTS FOR A FEW SECONDS, THEN YOU NEED TO RELEASE THE JOYSTICK IN ORDER TO RECHARGE ITS POWER.
*START GAME OPEN HANGAR

of R-Type, but nothing. With amazement and wonder I saw that the ship, when I held the joystick in the back position, came wrapped in a shield that protected me from bumps and enemy gunfire.

At the same time a line of energy placed under the screen went down and there I understood the absolute stroke of genius, which is that to get ahead in this game you have to study more the defensive phase than the offensive one. In fact the only way to survive is to use the shield not only as a defense, but also to collect enemy bullets, to be used later against them. In a few minutes the game takes over, you can't tear yourself away from it!

We have only one paltry life at our disposal but it's amazing how, by planning the use of the shield with the positioning of the ship, we can proceed even when the game becomes fast. You tend to forget that we have a few shots to shoot, but when you remember it is because the situation has become really tough. Graphically the game is very well done, with a parallax of stars that flow below us, while the enemy ships have definitely taken as an example those of Galaxian and Galaga. Nice is the ability to change, in the main menu, the color of the shield and the shape of our ship, with profiles that are unlocked upon reaching certain scores.

The music in the game has a techno edge and is addictive, while the sound FX fits well with it, never giving the idea of taking away a voice. Ultimately a game outside the logic of the genre that will appeal, for this very reason, even to those who do not particularly like to shoot everything that moves on the screen.

by **Michele Novarina**

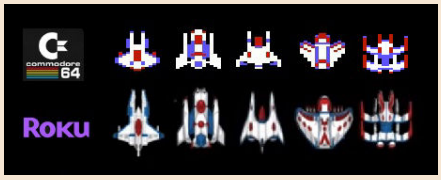
OUR FINAL SCORE

» Gameplay 85%

Initially you have to get familiar with the game mechanics completely reversed than usual.

» Longevity 95%

Once you understand the game mechanics, it's almost impossible to disengage from it.



NEW GAME

MEGA TURRICAN DIRECTOR'S CUT

Year: 1994-2021
Editor: Data East/Strictly Limited
Genre: Platform
Platform: Sega Megadrive
Website: https://store.strictlylimitedgames.com/products/mega-turrican-collection-mega-drive-preorder

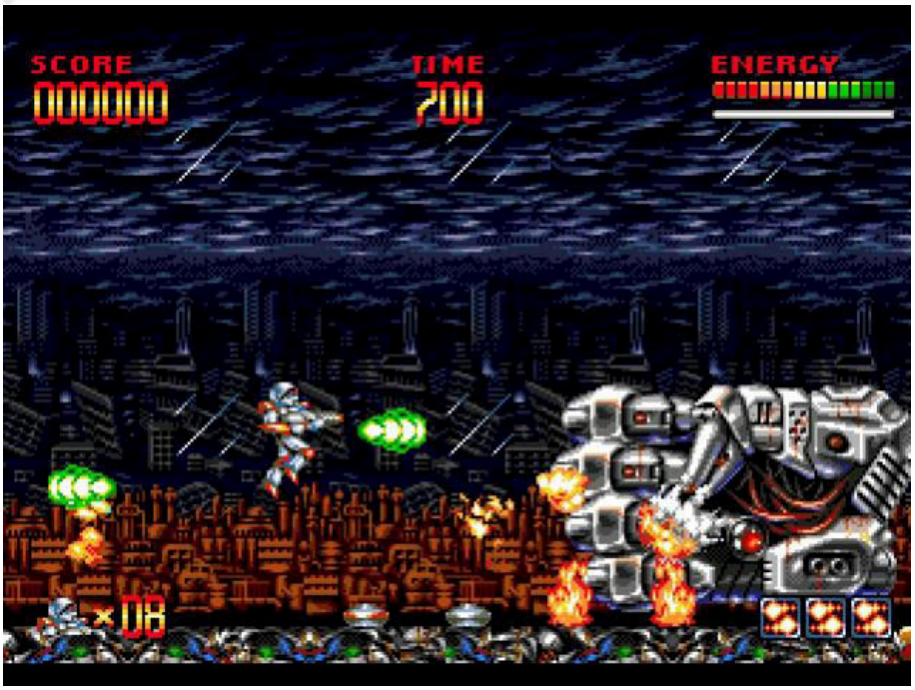
Very recently has been released the Director's Cut version of this classic on Sega Megadrive.

What does it contain? The original version of the game, the score Attack version (where you have to fight with time and make the best possible score) and the Director's Cut version that implements some graphic improvements. All this for 50 euros, a penny more... a penny less.

When I was a kid I loved anything Turrican branded, I would spend hours in front of the Amiga, lost in its levels and colorful graphics. The C64 version was pretty spectacular too.

Unlike today's games where what you see is what you get, back in the day, whether it was 8-bit or 16-bit, you needed to open up your imagination and being part of the wonderful universe on which the hero's exploits of Turrican unfolded was incredible. I was really immersing myself there.

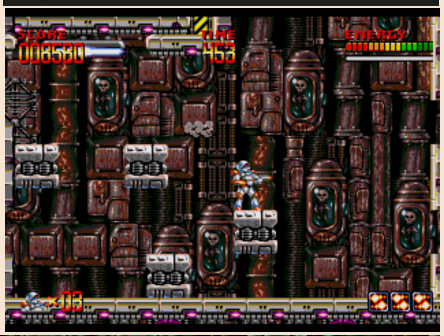
The game is basically a fairly linear run & gun, but there was something about the Turrican franchise that captured my imagination to the point where I found myself crafting some stories for a modern role-playing game series (GURPS, who knows?). Mega Turrican came at a time when



OUR FINAL SCORE

» Gameplay 80%
It works well until you jump. In some places it will make you lose temper.

» Longevità 70%
Several very large areas to explore with three levels of difficulty, but there are ultra easy situations and situations where the difficulty ramps up exaggeratedly. There is no real balanced curve.



the Amiga was losing ground and the Megadrive was in the difficult moment between the 32X failure and the launch of the Saturn, but as soon as it arrived I got it. Damn! Turrican and the Sega Megadrive, who knows what comes out!!!

It's hard to put into words my first impression. I was not satisfied at all. The magic was gone. The old power-ups and fancy game layouts were still there, the graphics were colorful even if repetitive, but something was missing.... The magic of previous games on other platforms was missing. Now I would call it differently and I would say that there was a lack of acceptable gameplay. It was made the minimal effort to catch the fans of the series, but without adding that extra touch to make it a masterpiece worthy of its name. It was terrifying the difficulty of jumping on some platforms; whoever programmed Mega Turrican has deliberately decided to make it extremely difficult. On many occasions you have to calculate each millimeter, pixel by pixel... otherwise you are dead!

This is not unfamiliar to platformers, but in this version of the game it is unbalanced and contrasts with the perfect gameplay of the chapters on Amiga or C64.

Sure there are a lot of upgrades, weapons and many levels but the action remains repetitive and the game never seems to take off.

Forget Chris Huelsbeck's music, here there's a soundtrack that you're unlikely to remember.

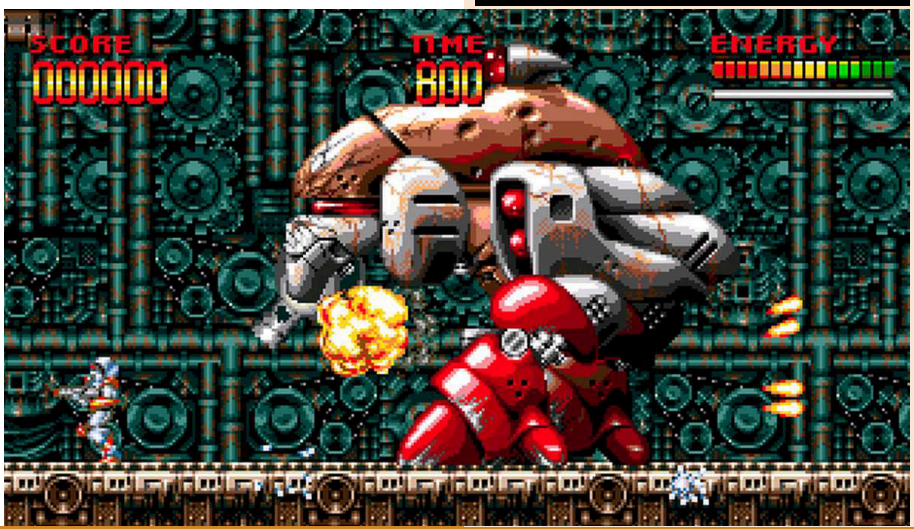
Too bad really. It could have been a new masterpiece. It is a good platformer but for me it is not Turrican.

This new edition adds two new game systems.

The Director's Cut version adds something, but very little.

It lacked commitment...

by **Giampaolo Moraschi**

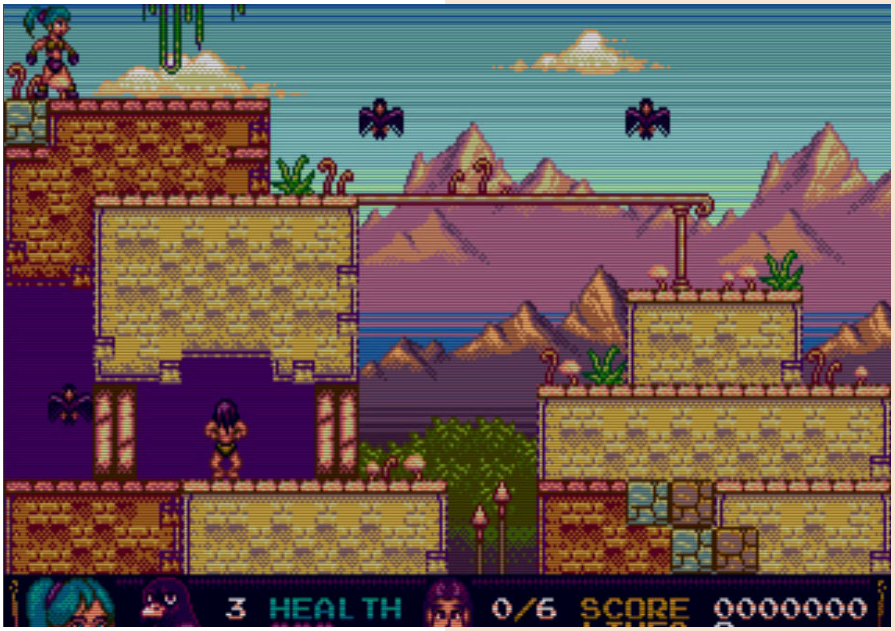


NEW GAME

EYRA, THE CROW MAIDEN

Year: 2022
Editor: Second Dimension
Genre: Platform
Platform: Sega Megadrive, Nes e Super Nintendo
Test on: Sega Megadrive
Website: https://second-dimension.com/store/eyra-the-crow-maiden-sega

Eyra is a young priestess of a barbarian tribe. Her friends and family have been kidnapped by the Infernal Marauder, an evil warrior sorcerer from a distant land. The young Amazon will set out to find the survivors against terrible hordes in the company of her sword and her brave trained raven.



It is a classic platform game where you have to jump, avoid traps and eliminate monsters, plus we have the task of freeing the prisoners in the levels and collecting some objects (represented by crows) that will enhance our main weapon, donate treasures or restore our health.

Second Dimension is working really well in recent times. Not simple homebrews with no rhyme or reason, but games with a good storyline and a great technical aspect. Often accompanied by well-made packaging and made with care manuals.

Eyra is also part of this type of product.

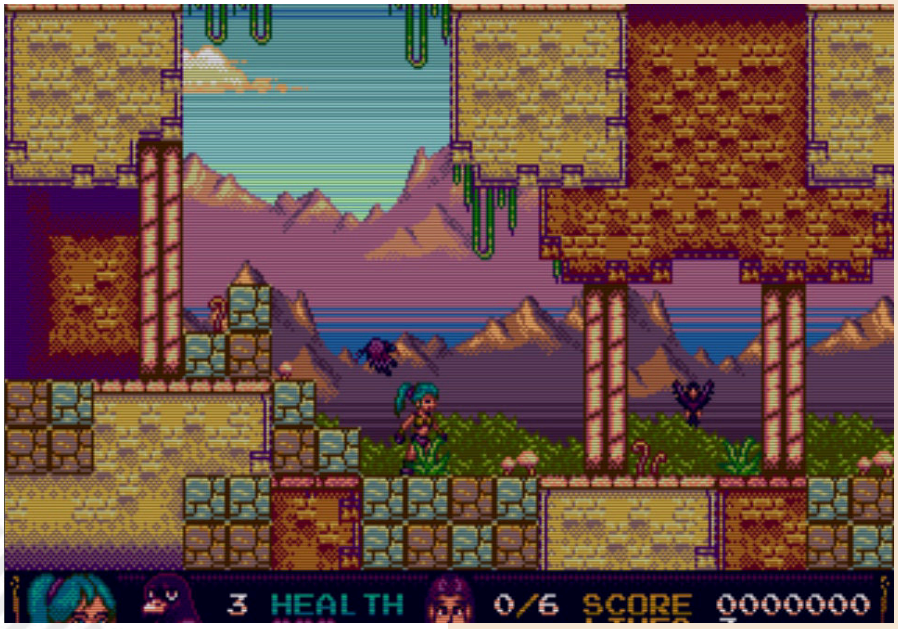
It's graphically pleasant and well made, it reminds me a lot the European Sega Megadrive productions and I'm sure that such a product would run worthily on Amiga too.

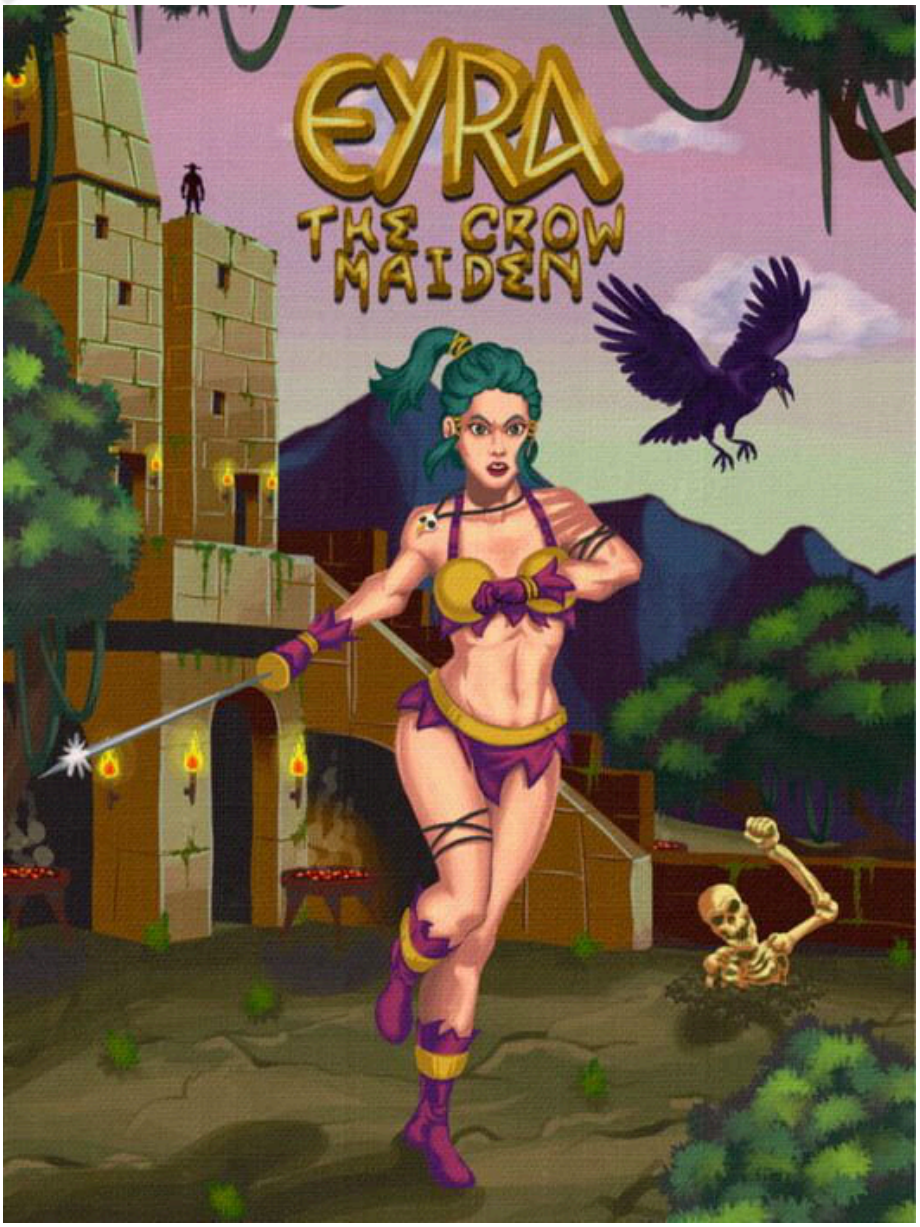
The soundtrack is repetitive but not annoying.

However, there is a problem...

It's a very flat game. Kill monsters, jump over obstacles, retrieve prisoners.

Not that it's a bad thing, but in the

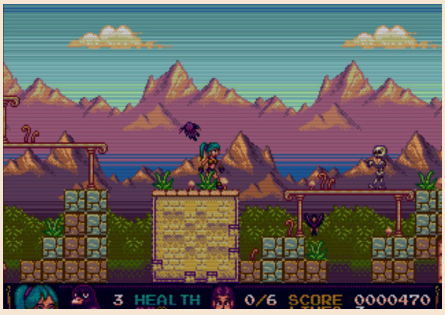




OUR FINAL SCORE

» Gameplay 70%
The classic platformer. One key jumps, one key hits, one key throws the crow. Nothing more.ù.

» Longevity 70%
It may seem repetitive in the long run.



long run it gets repetitive and can even be boring.

Available as a digital download and soon in a physical version..

I recommend you to test the demo present in the site that I put in the details. If you like it you can decide to buy the digital download (as I did) or push yourself to pre-order the physical version that seems really well made.

Either way, it's a more than adequate product.

by **Roberto Del Mar Pirazzini**

NEW GAME

KNIGHT GUY IN LOW RES WORLD - CASTLE DAYS

Year: 2021
Developer: Vladimir Zuniga
Genere: Platform
Platform: Atari 7800
Website: https://vhzcgames.com/knightguycd7800.php

Story: linear, an evil dragon has kidnapped our pet.
Gameplay: classic and fun with 100 levels to tackle full of traps and enemies.

A beautiful adventure for Atari's 8-bit console. A console that often went unnoticed but that found itself fighting the battle with two giants of its era, the NES and Master System.

As we said, the goal of the game is simple. Our knight must overcome 100 levels by avoiding all the dangers present, recovering weapons and healing potions and facing the end-of-level bosses present.

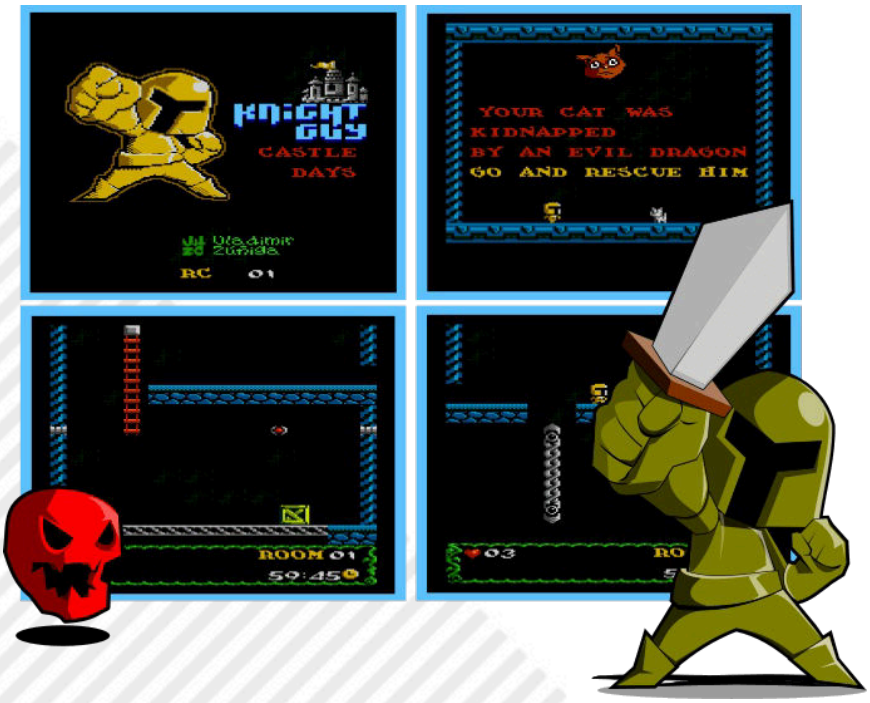
It was recently released also the physical version on cartridge, but if you want on the AtariAge site you can download the rom.

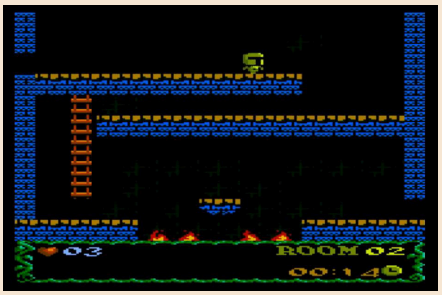
Personal opinion. Graphically simple but neat game, sound made only of effects and a robust playability.

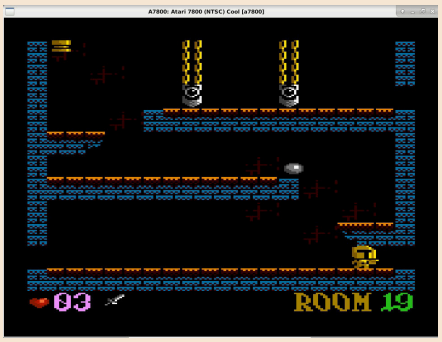
Recommended for lovers of the platformer/puzzle genre. It may not be a milestone but it brings a lot of fun.

You can buy the physical version from: https://atariage.com/store/index.php?l=product_detail&p=1262

by **Carlo Nithaiah Del Mar Pirazzini**







OUR FINAL SCORE

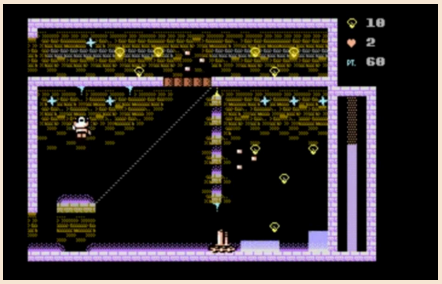
» Gameplay 80%
Simple and very eye-catching in level design.

» Longevity 80%
It's not easy and, in some places it's really maddening, but you'll find yourself replaying it over and over again.

NEW GAME

ROBOT JET ACTION

Year: 2022
Developer: Tomasz Mielnik e Kamil Wolnikoswski
Genre: Platform
Platform: Commodore 64
Website: https://carrion64.itch.io/robot-jet-action



Robot Jet Action is a classic fixed-screen platform game for our beloved Commodore 64/128.

It was made by two Polish super enthusiasts who run the C64Portal.pl portal (which I recommend you to visit).

In their new title the player will control a small robot with the joystick. By clicking on the fire button it will be able to activate the jetpack that will be used to reach all the objects to be collected on the screen.

Our little robot is the keeper of the trophies collected by millions of retrogamers in the world of past video games, but evil aliens/robots/villains are taking them away.

The game features beautiful HI RES graphics in ECM mode, nice animations and a really well-made soundtrack.

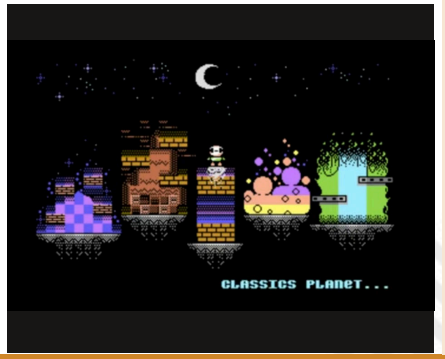
There are 35 levels, all of them really challenging. Not only we have to avoid our enemies but also all the obstacles in the level and we have to do it as fast as possible.

These types of titles stimulate me tremendously and lead me, even after dying over and over again, to replay the title.

A nice mix between BombJack and Manic Miner that is really worth playing. It's free!!!

Support developers for our retro systems.

by **Carlo Nithaiah Del Mar Pirazzini**



OUR FINAL SCORE

» Gameplay 85%
Immediate and without too much to think on how to move. The robot moves smoothly.

» Longevity 85%
As many as 35 levels and a good dose of difficulty that will keep you glued to the joystick.

SHONEN JUMP'S ONE PIECE

Year: 2004

Editor: Bandai/Dimps Corporation

Genre: Platform/Beat em up

Platform: Game Boy Advance

Finally mine! The platformer with fighting elements that I most wanted on Game Boy Advance.

I've always loved One Piece and Eiichiro Oda's drawings! And this anime, even if it's really long, never gets boring.

How many afternoons spent coming home from school to watch the episode on local TV. Always a unique thrill.

Moving on to the game, here we will guide Monkey D. Luffy (in Italy: Rubber) grappling with 12 bosses to face through 12 levels full of enemies and various dangers.

Colorful and well animated graphics. The sound is good too. It accompanies the adventure without ever bothering. The Audio and Visual Department is at the top.

It is a very simple and pleasant game to play which offers a medium-level challenge.

It will keep you busy for a long time.

Don't lose sight of it and try it out; in real hardware or emulation.

by **Barbara "Morgana" Murgida**

SHONEN JUMP'S
ONE PIECE
Press START
One Piece © 2005 Eiichiro Oda / Shueisha, Toei Animation.
©2004 - 2005 Bandai. All Rights Reserved.
Licensed by Nintendo

OUR FINAL SCORE

» Gameplay 80%
It's a pleasure to play with.

» Longevity 80%
Not too complicated nor too simple. A good compromise.

GREMLINS 2
THE NEW BATCH

Year: 1991

Editor: Sunsoft

Genre: Platform

Platform: Game Boy

In 1984 the first Gremlins, a cult film produced by Steven Spielberg and directed by Joe Dante, was released in America.

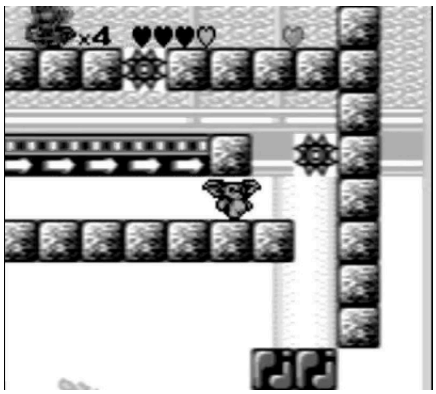
A masterpiece of the genre.
Two years later, the sequel came out.
A resounding flop.

As a fan of 80s/90s movies, since I was a young girl of 16, this Game Boy title could not be missed from my collection.

That said, it is a platformer graphically well cared where the little mogwai will have to make its way through the levels of the building where he's trapped; to do so he's equipped with weapons, more or less wacky, as a super tomato or a bow shoots flames.

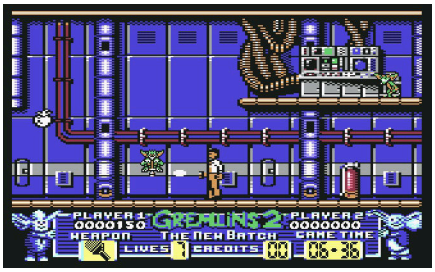
The goal is to get to the Gremlin control center and wipe out all the evil half-brothers inside.

I liked the game and I found a very cute Gizmo. It reminds me a lot of the "Furby" , those funny stuffed toys that were sold in Italy in the 90s.



By the way did you guys like the movies?

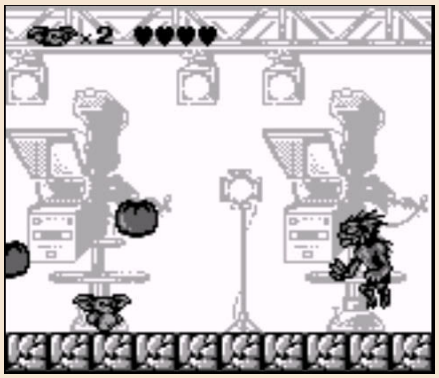
by **Barbara "Morgana" Murgida**



Other versions

There are several licensed versions of the game for numerous platforms. The versions for Amiga, Atari St, C64, MSX, Amstrad and ZX spectrum are basically side-scrolling platformers, not all of them very successful, with the main character Billy looking for Gizmo, while the Sunsoft version made for Nintendo NES is an action game with a top-down view, graphically nice and playable.

by **Carlo N. Del Mar Pirazzini**



OUR FINAL SCORE

» Gameplay 80%
A fun and easy to play game.

» Longevity 80%
Sometimes it's difficult but not frustrating. A nice walkthrough title on the Game Boy.

CRISIS FORCE

Year: 1991
Editor: Konami
Genre: Shoot'em up
Platform: Famicom/Nintendo NES

By 1991 it was clear that the days of Nintendo's glorious 8 Bit platform were numbered.

Despite the quality of the games of the rival Pc Engine and Megadrive and the arrival of the new Super Nintendo, some developers did not seem so convinced to surrender to the age of the small console.

Konami, Capcom and Enix were still making great titles for the Nes.

Crisis Force is an example of them. One of the greatest vertical scrolling shoot em ups ever released on this console.

The story is typical: 199X, Tokyo invaded by a robotic alien force. A boy and his girlfriend take off in their Falcon fighter planes to stop the threat.

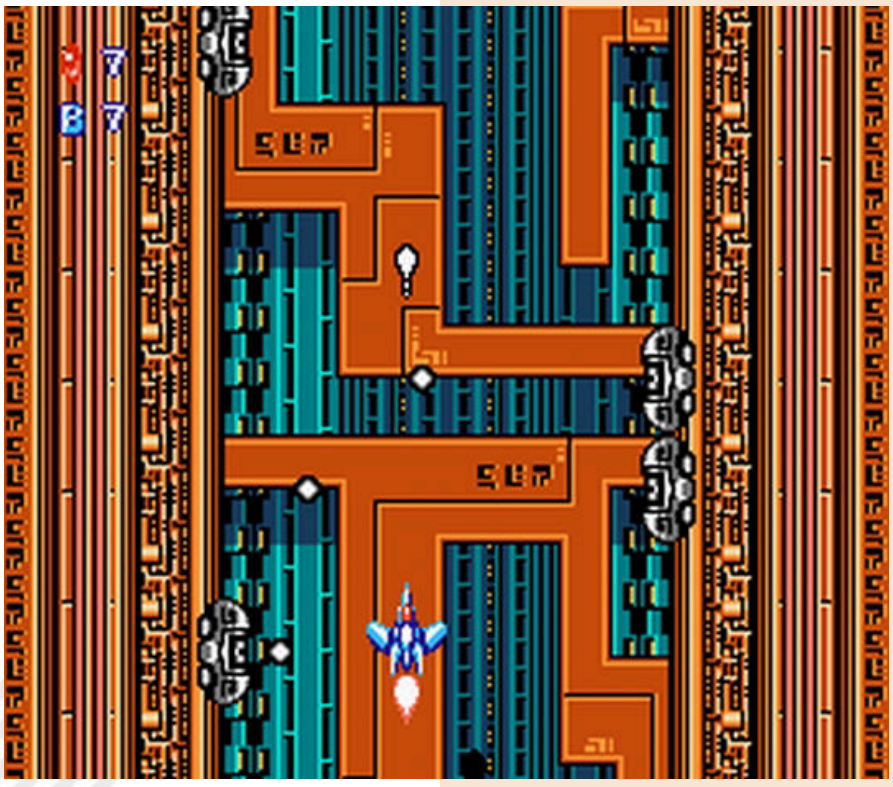
The game has classic power-ups, two types of laser beam power-ups and a blue icon that, when collected in quantity, makes the ship invincible for a short period.

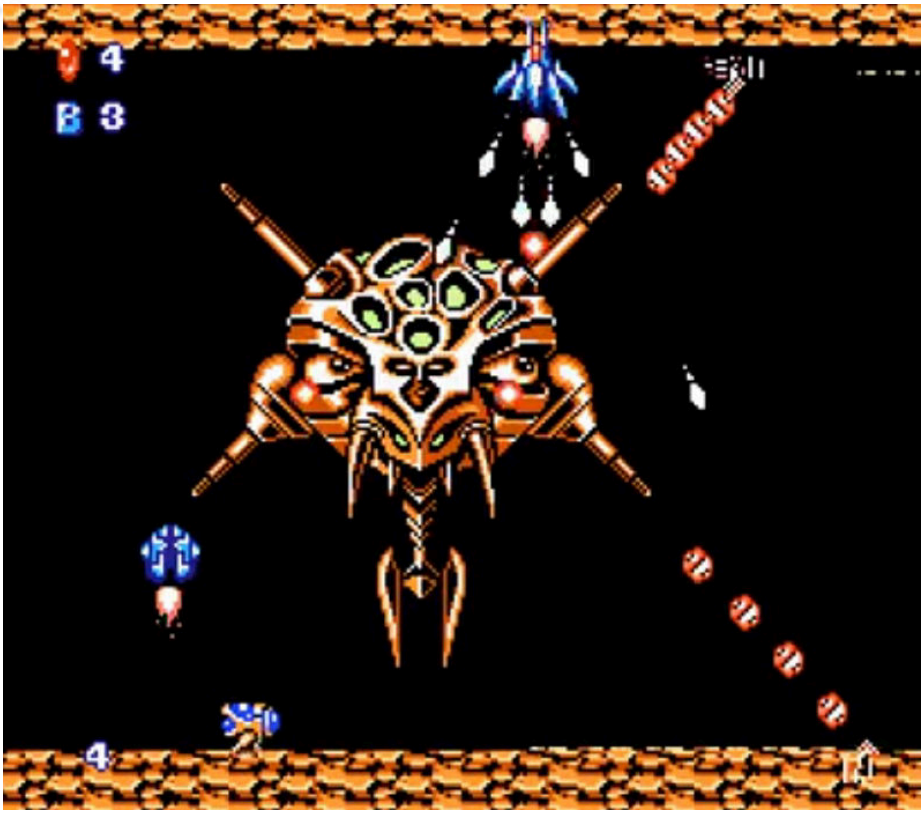
If you are playing doubles, by taking this blue icon you can combine with your playing partners and make the pair invincible.

Our ship can also be transformed during the game. Pressing the A key will change it from the standard version to the form with a cannon on the tail part. Pressing the A key again will transform it into a ship with four combat turrets on the sides.

The graphics in this game are PHENOMENAL! Crisis Force was often called "the AXELAY of FAMICOM" by the magazines of the time, and rightly so I might add.

Konami achieved a level of detail and technique worthy of 16-bit, in terms of multiple scrolling backgrounds and large sprites. Custom chips inside the cartridge made this feat possible. The first time I saw the game in action I was shocked. I had never seen anything like it on the NES/Famicom. The same goes for the audio department that,



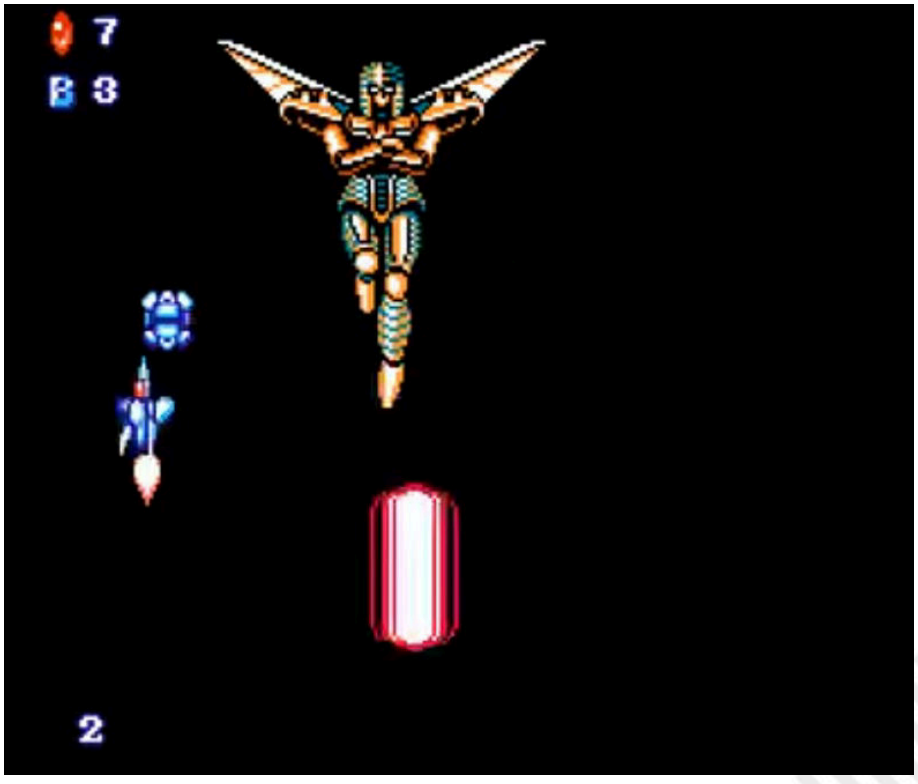


in perfect Konami style, is impeccable and engaging.

It's a long game with a well-balanced difficulty curve. It's certain that you have to try hard to advance through the levels with the right power-ups, but it's not a titanic feat.

Crisis Force is an exceptional shooter, ranked among the best NES games of the genre. It went unnoticed on the European market but had an incredible success in Japan. Curiously, it never reached American soil.
Try it out!

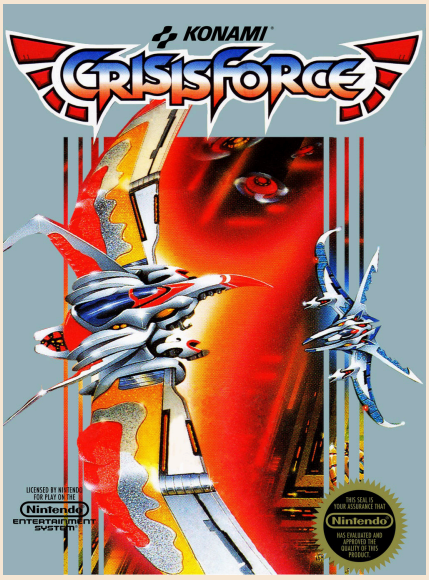
by **Roberto Del Mar Pirazzini**

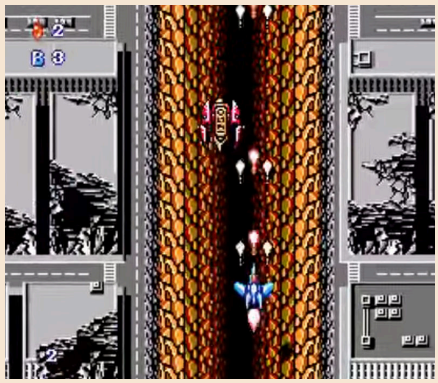


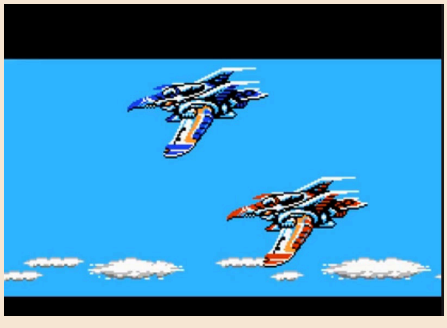
OUR FINAL SCORE

» Gameplay 95%
It is a Konami title. Simple in gameplay but fun. The ability to transform ships during the game makes it appealing.

» Longevity 90%
Seven levels are a nice challenge. The difficulty level is high but enjoyable.







CLASSIC KONG

Year: 2012
Developer: Bubble zap
Genre: Platform
Platform: Super Nintendo

It's a rainy Friday night and I'm obsessively playing Classic Kong that Nith handed me to test out.

I don't know the golden era of these games (I'm very young), but thanks to the passion of my father and the friendship of "uncle" Nith, I got lost in them. Literally in love.

I was saying I'm obsessively playing this Bubble Zap game for the Super Nintendo because I'm stubborn and I love this title.

It's been a long time since I've taken over the little Italian-American plumber, and this is the first time I've put my hand on its most ancestral incarnation.

An ugly dirty ape who steals girls and takes them to the highest part of the level. Poor Mario! He's always been unlucky in love.

The game system is the one you all know (I actually just recently, but I've been reading up on it) and it's brilliant, simple and punishing if you're not careful.

This Super Nintendo version is really enjoyable. Bright and nice graphics. That uses very well the colors of the console and seems to come straight out of the 90s, the golden age of the 16-bit machine.

The tagline "How high can you go?" we find between levels hints at how mesmerizing and playable this title is. We always want to get to see more and tackle the same levels with the difficulty rate increasing at a dizzying pace.

I was glued to it for hours and I must have cursed all the languages I know every time poor Mario got killed. But I'm still here.
A quality game, it doesn't look like a last generation homebrew but a title to put in the showcase with the other gems for this console.

I have tried the Rom on my Super Nintendo Mini, but Nith tells me that it runs without problems also on Everdrive and that on the web you can also find flash cartridges with packaging and game manual.
Happy research.

by **Ingrid Poggiali**

OUR FINAL SCORE

» Gamepley 90%
You have to be precise in jumping, hitting and getting the timing right... but the controls are perfect!

» Longevity 95%
The grade I gave speaks for itself.

PROHIBITION

Year: 1987
Developer: Infogrames
Genre: Shooter
Platform: Commodore 64

This game brought back some vague memories of a particular first-person shooter with crosshairs, underrated and never converted for any home machine: I'm talking about Empyre City or more commonly Street Fight (not to be confused with the legendary fighting game), set in the America of the gangsters of the 20's. But something, the Commodore 64 has brought home, although not at the levels of the latter, that is its clone, named Prohibition and signed by Infogrames.
As the name implies, the setting is among the infamous American neighborhoods and precisely during the years of Prohibition. The objective of the game is to kill every gangster that crosses our path, or rather, look, since we'll use the crosshairs of a gun. At regular intervals one of the gangsters will appear at the corners, at windows, on roofs and even in a manhole! All we have to do is shoot him. Up to here it might seem like a child's play (in every sense)!

The screen is not fixed as in the old arcade booths with infrared guns, but scrolling, and the direction to take is dictated by the arrows that appear at the bottom of the screen. We will have only five seconds to find the enemy and fill him with lead. Fortunately, the five seconds can be renewed by pressing the spacebar, which will make us jump to the ground to avoid the first enemy shot, but this will cost us energy, represented by some bullets.

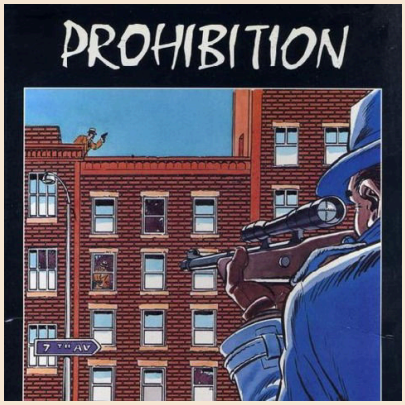
The sound of the game is very reminiscent of that of the gangster movies of the 20s, with music that was probably very popular at the time... Like nowadays summer hits.

And what about the gameplay? Moving the crosshairs of a gun is not optimal with a joystick or buttons, although the satisfaction of having taken out the enemy one second before time expires is really enjoyable.
As I said, the game is a scrolling one and you have to search the entire neighborhood for even more protected enemies: from the most unthinkable shelters to hostages! As in Empire City, there's a bandit who shields himself with a beautiful woman; this is surely the scene that impressed me the most in the original coin op.
It's one of those games that, even once you get to the end, you are tempted to play every now and then as a panacea especially after a stressful day at work to let off steam. While I was trying it, I was afraid it might be a little bit repetitive but it's not, there are small changes in enemies and environments going forward.
Loading from a cassette is not even long, just enough time for a coffee or tea or maybe a nice drink even though it would not be very suitable given the title.

Last but not least: Infogrames is a brand that has always produced good titles, even if not many have had the spotlight and, like any game that has received little attention, Prohibition deserves a second chance.

We always look for the bright side, because even in the thickest darkness there is a glimmer of light! After almost two years of pandemic and a significant slowness in recoveries, I wish those in quarantine a good recovery and a return to normal life.

by **Daniele Brahimi**



OUR FINAL SCORE

» Gameplay 70%
Killing gangsters with one second to go is satisfying!

» Longevity 60%
Not long, but you'll play it game every now and then.

Year: 2018
Platform: Discord
Link server: https://discord.io/retro_multiplayer
Link Social: https://linktr.ee/retromultiplayer

The spread of social networks has undoubtedly rekindled many passions of our childhood and among them there is also the one for retro-games.
In fact, the web is full of pages, groups and forums dedicated to the masterpieces of the arcades and the various home consoles.

But while most of these realities are limited to resurface memories and anecdotes of these gems of the past or at most explain how to replay them in single player, slowly made its way Retro Multiplayer. A community that aims to replay those titles in versus or cooperative mode, just like when we went to the arcades or at the home of our friends.

Retro Multiplayer was born about four years ago, when a small group of video game enthusiasts created a discord server and began with tenacity to experiment how to replay arcade titles in multiplayer online.
Test after test, success after success, that small group began to expand, finding many other users who were fascinated and involved in the project.

Today the discord server has more than five hundred members and has been joined by a Facebook group and page, an Instagram and Twitter profile and a Youtube and Twitch channel on which have been uploaded more than one hundred videos including tutorials and event nights.

In fact, Retro Multiplayer users, in addition to meeting in the evening to play together the retrogames, since 2020 have begun to organize themed events broadcasted regularly on various social channels.
Among these evenings there are tournaments, parties, music quizzes, retro interviews to the most popular Italian retrocomputing and retrogaming realities and of course cooperative games.

In addition, a few days ago the Retro Multiplayer team has released a free console with the various emulators set in every detail, so as to make the experience of online multiplayer extremely user friendly.

You'll be able to play titles ranging from classic arcade and 16-bit consoles to the Nintendo 64, PSP, Playstation, Dreamcast and Gamecube, as well as several PC milestones.

If you want to breathe again the atmosphere of arcade rooms and afternoons at friends' houses, you just need to enter the discord server by following the links found on this page.

You will meet a community of enthusiasts that has practically become a family that meets on a daily basis to share and relive together the endless passion for the magical world of retrogaming.

by **Querino Ialongo**

Some posters of events organized by the community

Before the closure of the issue, Giuseppe sent us another beautiful image, and it would be a crime not to share it with you. Artwork by **Giuseppe Mangini** inspired by the game **Knightmare**.

Taking a walk after lunch...

There was a time when I could never get enough information about computers, a time when I would have paid far more than I actually did buying magazines at the newsstand, manuals and guides at the bookstore, etc. All just to have the chance to look over the horizon and understand what it was hiding. I used to live in a small town near Verona, which is now a hub of many activities but, back then, too small to reunite a "critical mass": among all my friends there was not one who had the same computer as me. I had a C128, a friend had a Spectrum, another one had gone from VIC-20 to a PC based on an IBM 8088 processor, another was about to get his hands on an Amiga 500. All very nice but, in fact, our systems were not related to each other. As a result, despite all our attempts, no common project has ever blossomed. Unfortunately, despite what Thomas Merton said, each of us was, in fact, an island.

For years, in modern times, I have tried to go back to recover the "lost" opportunities, also thanks to the newfound love of many fans for retro-computing and the huge availability of material that then, as mentioned, was almost impossible to find, especially in Italian. Someone said that, if he would have had a computer, Einstein would perhaps have built a time machine; in my small way, if I had a couple more books, certainly I would not have built such a gimmick, but probably a videogame inspired by it. The "hunger" for knowledge was so strong that in recent times, before actually going back to programming on 8-bit computers, I had a series of recurring dreams that I later documented on the blog I collaborate with. Today those dreams are no more there, because in my reality RetroMagazine World exists as a wonderful place where I can experiment a thousand things, not only with my old (and still working) Commodore C128, but also with any 8 and/or 16-bit system I feel like working with. Software preservation, hardware info dissemination, creation of a structured and shared knowledge, stimulation of curiosity, integration, cooperation and much more..., this is what RetroMagazine World brings.

However, it is not the only possible world. Dozens of other equally stimulating hobbies knock on my door every day: from model making to language study, from comics (of all kinds) to Japanese animation. And so, even for me, things occasionally go out of focus. A few days ago, for example, I found myself asking a question to my colleagues in the editorial staff and I got the following answer: "Look, we talked about it in the magazine not long ago!". And so, like the sick people in "Johnny Mnemonic", am I too subject to information overload and their consumption only in pills? Could it be that the "hunger" I had has been satisfied to the point of making me lazy? We used to be hungrier in our past and that sharpened our wits. Now I better go out for a walk, even as a metaphor only. After regaining focus, I'm sure, the "hunger" will return as well.

Gianluca Girelli

Disclaimer

RetroMagazine World as an aperiodic magazine entirely ad-free is a non-profit project and falls off any commercial circuit. All the published material is produced by the respective authors and published thanks to their authorization.

RetroMagazine World is licensed under the terms of: **Attribution-NonCommercial-ShareAlike 4.0 International** (CC BY-NC-SA 4.0) https://creativecommons.org/licenses/by-nc-sa/4.0/

This is a human-readable summary of (and not a substitute for) the license. You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms. Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

RetroMagazine World-Eng
Year 3 - ISSUE 13 - FEBRUARY 2022

Chief Editor
Francesco Fiorentini
Managing Editor
David La Monaca
Editing Manager
Marco Pistorio
Web Management
Giorgio Balestrieri